

FILE ID**EXCDBG

M 4

EXCH
VO4-

20
74

20
74

20
20

20
20

20
73

20
20
00

```

EEEEEEEEE   XX      XX      CCCCCCCC   DDDDDDDD   BBBBBBBB   GGGGGGGG
EEEEEEEEE   XX      XX      CCCCCCCC   DDDDDDDD   BBBBBBBB   GGGGGGGG
EE           XX      XX      CC          DD          DD          BB          GG
EEEEEEEEE   XX      XX      CC          DD          DD          BBBBBBBB   GG
EEEEEEEEE   XX      XX      CC          DD          DD          BBBBBBBB   GG
EE           XX      XX      CC          DD          DD          BB          GG   GGGGGG
EE           XX      XX      CC          DD          DD          BB          GG   GGGGGG
EE           XX      XX      CC          DD          DD          BB          GG   GG
EE           XX      XX      CC          DD          DD          BB          GG   GG
EEEEEEEEE   XX      XX      CCCCCCCC   DDDDDDDD   BBBBBBBB   GGGGGG
EEEEEEEEE   XX      XX      CCCCCCCC   DDDDDDDD   BBBBBBBB   GGGGGG

```

The diagram illustrates the assembly of a DNA duplex from two types of monomers: L and S. The monomer L is represented by a vertical line with two horizontal bars extending to the left, while the monomer S is represented by a vertical line with two horizontal bars extending to the right. The assembly process starts with a single L monomer at the bottom left. It then adds S monomers to its right, forming a chain of alternating L and S monomers. This chain then begins to fold back upon itself, with the S monomers on the left side of the chain adding to the right side, creating a loop. This continues until the entire chain forms a complete circular structure, representing a double-stranded DNA molecule.

```

1 0001 0 MODULE exch$dbg
2 0002 0 (
3 0003 0     IDENT = 'V04-000'
4 0004 0     ADDRESSING_MODE (EXTERNAL=LONG_RELATIVE, NONEXTERNAL=WORD_RELATIVE)
5 0005 0 )
6 0006 1 BEGIN
7 0007 1 ****
8 0008 1 *
9 0009 1 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 *   ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 *
14 0014 1 *   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 *   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 *   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 *   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 *   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 *   TRANSFERRED.
20 0020 1 *
21 0021 1 *   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 *   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 *   CORPORATION.
24 0024 1 *
25 0025 1 *   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 *   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 ****
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY: EXCHANGE - Foreign volume interchange facility
33 0033 1
34 0034 1 ABSTRACT: Miscellaneous debug and trace routines
35 0035 1
36 0036 1 ENVIRONMENT: VAX/VMS User mode
37 0037 1
38 0038 1 AUTHOR: CW Hobbs           CREATION DATE: 16-July-1982
39 0039 1
40 0040 1 MODIFIED BY:
41 0041 1
42 0042 1 V03-002 CWH3002      CW Hobbs          12-Apr-1984
43 0043 1 Add routines to dump FAB and NAM.
44 0044 1
45 0045 1
46 0046 1 --
```

```
48 0047 1 ; Include files:  
49 0048 1 ;  
50 0049 1 MACRO $module_name_string = 'exch$dbg' %;      ! The require file needs to know our module name  
51 0050 1 REQUIRE 'SRC$!EXCREQ'  
52 0051 1 ;  
53 0148 1 ;  
54 0149 1 PSELECT  
55 0150 1     OWN = exch$dbg_own (WRITE)  
56 0151 1 ;  
57 0152 1 ;  
58 0153 1 %SBTTL 'Module table of contents'  
59 0154 1 ;  
60 0155 1 ! Module table of contents:  
61 0156 1 ;  
62 0157 1 FORWARD ROUTINE  
63 0158 1     exch$dbg_dump_dosv : NOVALUE,          ! Format the contents of a DOS11$ block  
64 0159 1     exch$dbg_fab_dev_print,           ! Print device characteristics from FAB info  
65 0160 1     exch$dbg_fab_dump,                ! Hex dump of a fab  
66 0161 1     exch$dbg_fab_name_print,         ! Print the name information attached to a FAB  
67 0162 1     exch$dbg_nam_dump,              ! Hex dump of a nam  
68 0163 1     exch$dbg_util_print_devchar    ! Explode a DEV device characteristics field (e.g. FABSL_DEV  
69 0164 1 ;  
70 0165 1 ;  
71 0166 1 ! EXCHANGE facility routines  
72 0167 1 ;  
73 0168 1 EXTERNAL ROUTINE  
74 0169 1     exch$dos11_expand_filename,        ! Expand filename info into a context block  
75 0170 1     exch$util_dos11ctx_allocate,       ! Grab a context block  
76 0171 1     exch$util_dos11ctx_release : NOVALUE ! Return the context block  
77 0172 1 ;  
78 0173 1 ;  
79 0174 1 ! Bound declarations:  
80 0175 1 ;  
81 0176 1 !BIND  
82 0177 1 ;
```

```
84      0178 1 GLOBAL ROUTINE exch$dbg_dump_dosv (volb : $ref_bblock, flags) : NOVALUE =
85      0179 2 BEGIN
86      0180 2 $dbgtrc_prefix ('dbg_dump_dosv>');
87      0181 2 LOCAL
88      0182 2   ctx : $ref_bblock,
89      0183 2   ent : $ref_bblock,
90      0184 2   lcl : $bblock [4],
91      0185 2   first
92      0186 2   ;
93      0187 2   ;
94      0188 2 BIND
95      0189 2   dosv = volb [volb$a_vfmt_specific] : $ref_bblock
96      0190 2   ;
97      0191 2   ;
98      0192 2 BUILTIN
99      0193 2   ACTUALCOUNT
100     0194 2   ;
101     0195 2   ;
102     0196 2 IF ACTUALCOUNT () EQL 2
103     0197 2 THEN
104     0198 2   lcl [0,0,32,0] = .flags
105     0199 2 ELSE
106     0200 2   lcl [0,0,32,0] = -1;
107     0201 2 IF .lcl [dmpdsv$v_volb]
108     0202 2 THEN
109     0203 2   Sprint_fao ('  dump_dosv> entry = volb !XL, dosv !XL', .volb, .dosv);
110     0204 2   $block_check (2, .volb, volb, 623);
111     0205 2   $block_check (2, .dosv, dos11, 624);
112     0206 2 IF .lcl [dmpdsv$v_status]
113     0207 2 THEN
114     0208 2   Sprint_fao ('  dump_dosv> status - posvalid !UL, bot !UL, tapemark !UL, eot !UL, dir !UL, curmrkd !UL'
115     0209 2   .dosv [dos11$v_position_valid], .dosv [dos11$v_beg_of_tape], .dosv [dos11$v_tape_mark],
116     0210 2   .dosv [dos11$v_end_of_tape], .dosv [dos11$v_directory], .dosv [dos11$v_current_marked]);
117     0211 2 IF .lcl [dmpdsv$v_position]
118     0212 2 THEN
119     0213 2   Sprint_fao ('  dump_dosv> .dos11$a_current_entry !XL .dos11$l_current_file !UL',
120     0214 2   .dosv [dos11$a_current_entry], .dosv [dos11$l_current_file]);
121     0215 2 IF .lcl [dmpdsv$v_entries]
122     0216 2 THEN
123     0217 2 BEGIN
124     0218 2   Sprint_fao ('  dump_dosv> .dos11$a_current_entry !XL .dos11$l_current_file !UL',
125     0219 2   .dosv [dos11$a_current_entry], .dosv [dos11$l_current_file]);
126     0220 2 IF .lcl [dmpdsv$v_entries]
127     0221 2 THEN
128     0222 2 BEGIN
129     0223 3   ! If the pointer points to the queue header, we have reached the end of the list
130     0224 3   !
131     0225 3   ent = .dosv [dos11$a_entry.flink];
132     0226 3   ctx = exch$util_dos11Tctx_allocate (.volb, 0);      ! Grab a context block for the loop
133     0227 3   first = true;
134     0228 3   WHILE .ent NEQ dosv [dos11$q_entry_header]
135     0229 3   DO
136     0230 3   BEGIN
137     0231 3   IF .first
138     0232 4   BEGIN
139     0233 4   IF .first
```

```

141      0235 4 THEN
142      0236 5 BEGIN
143      0237 5 $print_lit ('    dump_dosv> Num   Entry   Flink   Blink   Name           Block Val');
144      P 0238 5 $print_fao ('    dump_dosv> !3UL !XL !XL !XL !13AF !5UL !UL',
145      P 0239 5         9999, dosv [dos11$g_entry_header], dosv [dos11$a_entry_flink], .dosv [dos11$a_entry_bli
146      0240 5             12, UPLIT BYTE ('queue-header'), 999999, 99);
147      0241 4 END;
148      0242 4 first = false;
149      0243 4
150      0244 4 CH$MOVE (dos11ctx$g_entry, ent [dos11ent$L_filename_1], ctx [dos11ctx$g_entry]);
151      0245 4 exch$dos11_expand_filename (.ctx);
152      0246 4
153      P 0247 4 $print_fao ('    dump_dosv> !3UL !XL !XL !XL !13AF !5UL !UL',
154      P 0248 4         .ent [dos11ent$w_file_number], .ent .ent [dos11ent$a_flink], .ent [dos11ent$a_blin
155      P 0249 4         .ctx [dos11ctx$g_exp_fullname_len], ctx [dos11ctx$g_exp_fullname],
156      0250 4             .ent [dos11ent$w_blocks], .ent [dos11ent$v_blocks_valid]);
157      0251 4
158      0252 4     ent = .ent [dos11ent$a_flink];          ! Move to the next
159      0253 3     END;
160      0254 3
161      0255 3 exch$util_dos11ctx_release (.ctx);          ! Return the context block
162      0256 2 END;
163      0257 2
164      0258 2 RETURN;
165      0259 1 END;

```

.TITLE EXCH\$DBG Conditionally assembled debugging routines

.IDENT \V04-000\

.PSECT EXCH\$DBG_PLIT,NOWRT,2

```

20 3E 76 73 6F 64 5F 70 6D 75 64 20 20 20 00000 P.AAB: .ASCII \ dump_dosv> entry - volb !XL, dosv !
58 21 20 62 6C 6F 21 20 76 2D 73 20 6F 64 20 00 4C 58 00028
58 21 20 76 2D 73 20 6F 64 20 00 2C 4C 0001E
58 21 20 76 2D 73 20 6F 64 20 00 6E 65 0000F
58 21 20 76 2D 73 20 6F 64 20 00 20 20 0000F
58 21 20 76 2D 73 20 6F 64 20 00 4C 58 00028
58 21 20 76 2D 73 20 6F 64 20 00 010E002A 0002C P.AAA: .ASCII \XL\<0><0>
58 21 20 76 2D 73 20 6F 64 20 00 00000000 00030 P.AAA: .LONG 17694762
58 21 20 76 2D 73 20 6F 64 20 00 010E002A 00030 P.AAA: .ADDRESS P.AAB
20 3E 76 73 6F 64 5F 70 6D 75 64 20 20 20 00034 P.AAD: .ASCII \ dump_dosv> status - posvalid !UL, bo\
6C 61 76 73 6F 64 5F 70 20 2D 20 73 75 64 20 20 20 00043 P.AAD: .LONG 17694762
6C 61 76 73 6F 64 5F 70 20 2D 20 73 75 64 20 20 20 00052 P.AAD: .ADDRESS P.AAD
6B 72 61 6D 65 70 61 74 20 2C 4C 55 21 20 64 69 20 20 0005C P.AAD: .ASCII \t !UL, tapemark !UL, eot !UL, dir !UL, c\
20 2C 4C 55 21 20 74 6F 65 20 2C 4C 55 21 20 74 69 20 20 0006B P.AAD: .LONG 17694762
20 2C 4C 55 21 20 74 6F 65 20 2C 4C 55 21 20 74 69 20 20 0007A P.AAD: .ADDRESS P.AAD
20 2C 4C 55 21 20 74 6F 65 20 2C 4C 55 21 20 74 69 20 20 00084 P.AAD: .ASCII \urmrkd !UL\<0><0>
20 2C 4C 55 21 20 74 6F 65 20 2C 4C 55 21 20 74 69 20 20 00090 P.AAD: .LONG 17694762
20 2C 4C 55 21 20 74 6F 65 20 2C 4C 55 21 20 74 69 20 20 00094 P.AAD: .ADDRESS P.AAD
20 3E 76 73 6F 64 5F 70 6D 75 64 20 20 20 00098 P.AAF: .ASCII \ dump_dosv> .dos11$a_current_entry \
6E 65 72 72 75 63 5F 61 24 31 31 73 6F 64 2E 000A7 P.AAF: .LONG 17694792
6E 65 72 72 75 63 5F 61 24 31 31 73 6F 64 2E 000A7 P.AAF: .ADDRESS P.AAF
5F 6C 24 31 31 73 6F 64 2E 20 20 2C 4C 58 21 000B6 P.AAF: .ASCII \!XL, .dos11$l_current_file !UL\
21 20 20 65 6C 69 66 5F 74 6E 65 72 72 75 63 000C0 P.AAF: .LONG 17694792
21 20 20 65 6C 69 66 5F 74 6E 65 72 72 75 63 000CF P.AAF: .ADDRESS P.AAF
21 20 20 65 6C 69 66 5F 74 6E 65 72 72 75 63 000DE P.AAF: .ASCII \ dump_dosv> Num Entry Flink \
20 3E 76 73 6F 64 5F 70 6D 75 64 20 20 20 000E8 P.AAH: .LONG 17694792
20 3E 76 73 6F 64 5F 70 6D 75 64 20 20 20 000E8 P.AAH: .ADDRESS P.AAH
20 3E 76 73 6F 64 5F 70 6D 75 64 20 20 20 000E8 P.AAH: .ASCII \ dump_dosv> Num Entry Flink \

```


7E		50	OC	A7	9E	0006A	MOVAB	12(R7), R0
7E		60	01	05	EF	0006E	EXTZV	#5, #1, -(RO), -(SP)
7E		60	01	04	EF	00073	EXTZV	#4, #1, -(RO), -(SP)
7E		60	01	03	EF	00078	EXTZV	#3, #1, -(RO), -(SP)
7E		60	01	02	EF	0007D	EXTZV	#2, #1, -(RO), -(SP)
7E		60	01	01	EF	00082	EXTZV	#1, #1, -(RO), -(SP)
7E		60	01	00	EF	00087	EXTZV	#0, #1, -(RO), -(SP)
				00000	CF	9F 0008C	PUSHAB	P.AAC
		00000000G	EF		07	FB 00090	CALLS	#7, EXCH\$UTIL_FAO_BUFFER
					50	DD 00097	PUSHL	R0
		16	53	6B	01	FB 00099	CALLS	#1, LIB\$PUT_OUTPUT
					02	E1 0009C	BBC	#2, LCL, 5\$-
				OE	A7	DD 000A0	PUSHL	14(R7)
				1A	A7	DD 000A3	PUSHL	26(R7)
		00000000G	EF		00000	CF 9F 000A6	PUSHAB	P.AAE
					03	FB 000AA	CALLS	#3, EXCH\$UTIL_FAO_BUFFER
					50	DD 000B1	PUSHL	R0
		01	53	6B	01	FB 000B3	CALLS	#1, LIB\$PUT_OUTPUT
					03	E0 000B6	BBS	#3, LCL, 6\$-
				56	12	A7 DO 000BB	RET	
					7E	D4 000BF	MOVL	18(R7), ENT
		00000000G	EF		04	AC DD 000C1	CLRL	-(SP)
					02	FB 000C4	PUSHL	VOLB
				58	50	DO 000CB	CALLS	#2, EXCH\$UTIL_DOS11CTX_ALLOCATE
				59	01	DO 000CE	MOVL	R0, CTX
				5A	66	A8 9E 000D1	MOVAB	#1, FIRST
				50	12	A7 9E 000D5	MOVAB	102(R8), R10
				50	56	D1 000D9	CMPL	18(R7), R0
					74	13 000DC	BEQL	ENT, R0
				33	59	E9 000DE	BLBC	9\$
					00000	CF 9F 000E1	PUSHAB	FIRST, 8\$
				6B	01	FB 000E5	CALLS	P.AAG
				7E	63	8F 9A 000E8	MOVZBL	#1, LIB\$PUT_OUTPUT
					000F423F	8F DD 000EC	PUSHL	#99 - (SP)
					00000	CF 9F 000F2	PUSHAB	#999999
					0C	DD 000F6	PUSHL	P.AAK
				7E	12	A7 7D 000F8	MOVQ	#12
					12	A7 9F 000FC	PUSHAB	18(R7), -(SP)
				7E	270F	8F 3C 000FF	MOVZWL	18(R7)
					00000	CF 9F 00104	PUSHAB	#9999, -(SP)
		00000000G	EF		09	FB 00108	CALLS	P.AAI
					50	DD 0010F	PUSHL	#9, EXCH\$UTIL_FAO_BUFFER
				6B	01	FB 00111	CALLS	R0
					59	D4 00114	CLRL	#1, LIB\$PUT_OUTPUT
	3C	A8	08	A6	14	28 00116	FIRST	FIRST
					58	DD 0011C	MOVC3	#20, 8(ENT), 60(CTX)
		00000000G	EF		01	FB 0011E	PUSHL	CTX
	1A	A6			00	EF 00125	CALLS	#1, EXCH\$DOS11_EXPAND_FILENAME
				7E	16	A6 3C 0012B	EXTZV	#0, #1, 26(ENT), -(SP)
					5A	DD 0012F	MOVZWL	22(ENT), -(SP)
				7E	50	A8 DD 00131	PUSHL	R10
					66	7D 00134	MOVQ	80(CTX)
				7E	56	DD 00137	PUSHL	(ENT), -(SP)
					18	A6 3C 00139	MOVZWL	ENT
		00000000G	EF		00000	CF 9F 0013D	PUSHAB	24(ENT), -(SP)
					09	FB 00141	CALLS	P.AAL
								#9, EXCH\$UTIL_FAO_BUFFER

EXCH\$DBG
V04-000

Conditionally assembled debugging routines
Module table of contents

G 5
16-Sep-1984 00:45:26
14-Sep-1984 12:29:02

VAX-11 Bliss-32 V4.0-742
[EXCHNG.SRC]EXCDBG.B32:1

Page 7
(3)

EXC
V04

6B 56 00000000G EF	50 DD 00148 01 FB 0014A 66 D0 0014D 83 11 00150 58 DD 00152 9\$: 01 FB 00154 04 0015B	PUSHL R0 CALLS #1, LIB\$PUT_OUTPUT MOVL (ENT), ENT BRB 7\$ PUSHL CTX CALLS #1, EXCH\$UTIL_DOS11CTX_RELEASE RET		: 0252 0230 0255 0259
------------------------------	-------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------	--	-----------------------------------

: Routine Size: 348 bytes, Routine Base: EXCH\$DBG_CODE + 0000

: R

```

: 167      0260 1 GLOBAL ROUTINE exch$dbg_fab_dev_print (fabb : $ref_bblock) =    %SBTTL 'exch$dbg_fab_dev_print'
168      0261 2 BEGIN
169
170      0262 2 $dbgtrc_prefix ('fab_dev_print > ');
171
172      0263 2 OWN
173          0266 2 devbufsiz,
174          0267 2 devchar,
175          0268 2 devclass,
176          0269 2 devdepend,
177          0270 2 devnam      : $bblock [64],
178          0271 2 devnamlen,
179          0272 2 devtype,
180          0273 2 maxblock;
181
182      0275 2 OWN
183          0276 2 dev_item : VECTOR [22, LONG] INITIAL (
184              0277 2 WORD (dvi$ devbufsiz, 4),           LONG (devbufsiz,0),
185              0278 2 (dvi$_devchar^16 OR 4),           devchar,        0,
186              0279 2 (dvi$_devclass^16 OR 4),           devclass,       0,
187              0280 2 (dvi$_devdepend^16 OR 4),           devdepend,     0,
188              0281 2 (dvi$_fulldevnam^16 OR 64),         devnam,        devnamlen,
189              0282 2 (dvi$_devtype^16 OR 4),           devtype,       0,
190              0283 2 (dvi$_maxblock^16 OR 4),           maxblock,      0,
191              0284 2 0);
192
193      0286 2 LOCAL
194          0287 2 status,
195          0288 2 inp_desc : $desc_block,
196          0289 2 tmp_desc : $desc_block,
197          0290 2 nam : $ref_bblock;                  ! Pointer to the name block
198
199          0292 2 nam = .fabb [fab$l_nam];           ! Get pointer to the name block
200          0293 2 $stat_str_desc_init (inp_desc, .nam [nam$b_dev], .nam [nam$l_dev]); ! Input Device name address
201
202          0295 2 status = $getdviw (efn=0, devnam=inp_desc, itmlst=dev_item);
203
204          0296 2 IF NOT .status
205          0297 2 THEN
206              0298 3 BEGIN
207                  0299 3 $print_fao ('***** Error in $GETDVI for device "AS" *****', inp_desc);
208                  0300 3 status = ((.status AND %X'FFFFFFF8') OR 1);
209                  0301 3 SIGNAL ($warning_stat (status));
210                  0302 3 RETURN 0;
211                  0303 2 END;
212
213          0305 2 tmp_desc [dsc$w_length] = .devnamlen;
214          0306 2 tmp_desc [dsc$w_pointer] = devnam;
215          0307 2 $print_fao ('Getdvi for name "AS" resolved to device "AS"', inp_desc, tmp_desc);
216          0308 2 $print_fao ('Bufsiz = !UL, Maxblocks = !UL, Class = !XB, Type = !XB', .devbufsiz, .maxblock, .devclass, .
217
218          0309 3 BEGIN
219          0310 3 BIND
220              0311 3 dep = devdepend : $bblock,
221              0312 3 dev = devchar : $bblock;
222          P 0313 3 $print_fao ('Cylinders = !UL, Tracks = !UL, Sectors = !UL, DevChar = !XL',
223                  0314 3 .dep[0,16,16,0], .dep[0,8,8,0], .dep[0,0,8,0], .dev [0,0,32,0]);
224
225          0315 2 END;
226          0316 2

```

```
: 224 0317 2 exch$dbg_utl_print_devchar (.devchar);
: 225 0318 2
: 226 0319 2 RETURN 1;
: 227 0320 1 END;
```

.PSECT EXCH\$DBG_OWN,NOEXE,2

00000 DEVBUFSIZ:

.BLKB 4

00004 DEVCHAR:.BLKB 4

00008 DEVCLASS:

.BLKB 4

0000C DEVDEPEND:

.BLKB 4

00010 DEVNAM:.BLKB 64

00050 DEVNAMLEN:

.BLKB 4

00054 DEVTYPE:.BLKB 4

00058 MAXBLOCK:

.BLKB 4

0004 0008 0005C DEV_ITEM:

.WORD 8, 4

00000000' 00060 .ADDRESS DEVBUFSIZ

.LONG 0

00000000 00064 .LONG 131076

00020004 00068 .ADDRESS DEVCHAR

.LONG 0, 262148

00040004 00000000 00070 .ADDRESS DEVCLASS

.LONG 0, 655364

000A0004 00000000 0007C .ADDRESS DEVDEPEND

.LONG 0, 15204416

00E80040 00000000 00084 .ADDRESS DEVNAM, DEVNAMLEN

.LONG 393220

00000000' 00088 .ADDRESS DEVTYPE

.LONG 0, 1703940

00000000 00090 .ADDRESS MAXBLOCK

.LONG 0, 0

00060004 00098 .ADDRESS P.AAO

.ASCII ***** Error in \$GETDVI for device "!AS"\

001A0004 00000000 000A0 .LONG 17694767

00000000' 000A8 .ADDRESS P.AAO

.ASCII \Getdvi for name "!AS" resolved to device\

00000000 000AC .ASCII \!!AS"\<0><0>

.LONG 17694766

0010E002F 000204 P.AAN: .ADDRESS P.AAO

.ASCII \Bufsiz = !UL, Maxblocks = !UL, Class =\

00000000' 000208 P.AAQ: .ASCII \Getdvi for name "!AS" resolved to device\

00000000 00021B P.AAQ: .ASCII \!!AS"\<0><0>

0010E002E 00023C P.AAP: .ADDRESS P.AAO

.ASCII \Bufsiz = !UL, Maxblocks = !UL, Class =\

00000000' 000240 P.AAS: .LONG 17694766

00000000 000253 P.AAS: .ADDRESS P.AAO

.ASCII \Bufsiz = !UL, Maxblocks = !UL, Class =\

```
6E 69 20 72 6F 72 72 45 20 20 2A 2A 2A 2A 2A 001D4 P.AAO: .ASCII \***** Error in $GETDVI for device "!AS"\
65 64 20 72 6F 66 20 49 56 44 54 45 47 24 20 001E3
       22 53 41 21 22 20 65 63 69 76 001F2
       00 2A 2A 2A 2A 2A 20 20 001FC
       010E002F 00204 P.AAN: .ASCII \*****\<0>
       00000000' 00208 P.AAQ: .LONG 17694767
       00000000 0021B P.AAQ: .ADDRESS P.AAO
       0010E002E 0023C P.AAP: .ASCII \Getdvi for name "!AS" resolved to device\
       00000000' 00240 P.AAS: .ADDRESS P.AAO
       00000000 00253 P.AAS: .ASCII \!!AS"\<0><0>
```

```
20 20 2C 4C 55 21 20 3D 20 73 6B 63 6F 6C 62 78 61 4D 00253
4C 55 21 20 3D 20 73 73 61 6C 43 20 20 2C 00262
```

21 20 3D 20 65 70 79 54 20 20 2C 00 42 58 21 20 0026C	.ASCII \ !XB, Type = !XB\<0><0><0>	:
00 00 00 42 58 0027B		
010E0039 00280 P.AAR: .LONG 17694777		
00000000' 00284 P.AAU: .ADDRESS P.AAS		
4C 55 21 20 3D 20 73 72 65 64 6E 69 6C 79 43 00288 P.AAU: .ASCII \Cylinders = !UL, Tracks = !UL, Sectors =\		
2C 4C 55 21 20 3D 20 73 72 65 64 61 72 54 20 2C 00297		
6F 74 63 65 53 20 002A6		
3D 20 72 61 68 43 76 65 44 20 2C 4C 55 21 20 002B0	.ASCII \ !UL, DevChar = !XL\<0>	
00 4C 58 21 20 002BF		
010E003B 002C4 P.AAT: .LONG 17694779		
00000000' 002C8 P.AAU: .ADDRESS P.AAU		
	DEP= DEVDEPEND	
	DEV= DEVCHAR	
	.EXTRN SYSSGETDVIW	
	.PSECT EXCH\$DBG_CODE,NOWRT,2	
55 00000000G 00 9E 00002 003C 00000	.ENTRY EXCH\$DBG FAB DEV PRINT, Save R2,R3,R4,R5	: 0260
54 00000000G EF 9E 00009	MOVAB LIB\$PUT_OUTPUT, R5	
53 0000' CF 9E 00010	MOVAB EXCH\$UTIL_FAO_BUFFER, R4	
5E 50 04 AC D0 00018	MOVAB DEP, R3	
50 50 28 A0 D0 0001C	SUBL2 #16, SP	
0A AE 010E 8F B0 00020	MOVL FABB, R0	
08 AE 39 A0 9B 00026	MOVL 40(R0), NAM	
OC AE 44 A0 D0 0002B	MOVW #270, DESC+2	
	MOVZBW 57(NAM), DESC	
	MOVL 68(NAM), DESC+4	
	CLRQ -(SP)	
	CLRQ -(SP)	
	PUSHAB DEV_ITEM	
	PUSHAB INP-DESC	
	CLRQ -(SP)	
00000000G 00 08 FB 0003C	CALLS #8, SYSSGETDVIW	
52 25 0000' 50 D0 00043	MOVL R0, STATUS	
	BLBS STATUS, 1\$	
	PUSHAB INP DESC	
	PUSHAB P.AAN	
	CALLS #2, EXCH\$UTIL_FAO_BUFFER	
	PUSHL R0	
	CALLS #1, LIB\$PUT_OUTPUT	
50 52 65 01 FB 00055	BICL3 #7, STATUS,-R0	
52 50 07 CB 00058	BISL3 #1, R0, STÁTUS	
52 52 01 C9 0005C	BICB2 #7, STÁTUS2	
	PUSHL STÁTUS2	
00000000G 00 01 FB 00065	CALLS #1, LIB\$SIGNAL	
	BRB 2\$	
04 AE 44 A3 B0 0006E 1\$: 58 11 0006C	MOVW DEVNAMLEN, TMP_DESC	
04 AE 04 A3 9E 00072	MOVAB DEVNAM, TMP_DESC+4	
	PUSHL SP	
	PUSHAB INP DESC	
	PUSHAB P.AAP	
	CALLS #3, EXCH\$UTIL_FAO_BUFFER	
64 0000' 03 FB 00080	PUSHL R0	
	CALLS #1, LIB\$PUT_OUTPUT	
65 48 A3 DD 00088	PUSHL DEVTYPE	
FC A3 DD 00088	PUSHL DEVCLASS	

EXCH\$DBG
V04-000

Conditionally assembled debugging routines
exch\$dbg_fab_dev_print

K 5
16-Sep-1984 00:45:26
14-Sep-1984 12:29:02
VAX-11 Bliss-32 V4.0-742
[EXCHNG.SRC]EXCDBG.B32;1

Page 11
(4)

EXC
V04

	4C	A3	DD	0008E	PUSHL	MAXBLOCK	
	F4	A3	DD	00091	PUSHL	DEVBUFFSIZ	
		CF	9F	00094	PUSHAB	P.AAR	
64		05	FB	00098	CALLS	#5, EXCH\$UTIL_FA0_BUFFER	
		50	DD	0009B	PUSHL	R0	
65		01	FB	0009D	CALLS	#1, LIB\$PUT_OUTPUT	0314
		F8	A3	DD 000A0	PUSHL	DEV	
7E		63	9A	000A3	MOVZBL	DEP, -(SP)	
7E		01	A3	9A 000A6	MOVZBL	DEP+1, -(SP)	
7E		02	A3	3C 000AA	MOVZWL	DEP+2, -(SP)	
	0000'	CF	9F	000AE	PUSHAB	P.AAT	
64		05	FB	000B2	CALLS	#5, EXCH\$UTIL_FA0_BUFFER	
		50	DD	000B5	PUSHL	R0	
65		01	FB	000B7	CALLS	#1, LIB\$PUT_OUTPUT	
0000V	CF	F8	A3	DD 000BA	PUSHL	DEVCHAR	0317
	50		01	FB 000BD	CALLS	#1, EXCH\$DBG_UTL_PRINT_DEVCHAR	
			01	D0 000C2	MOVL	#1, R0	0319
			04	000C5	RET		
			50	D4 000C6 2\$:	CLRL	R0	
			04	000C8	RET		0320

: Routine Size: 201 bytes, Routine Base: EXCH\$DBG_CODE + 015C

```

229 0321 1 GLOBAL ROUTINE exch$dbg_fab_dump (fab : $ref_bblock) = %SBTTL 'exch$dbg_fab_dump'
230 0322 2 BEGIN
231 0323 2
232 0324 2 $dbgtrc_prefix ('fab_dump>');
233 0325 2
234 0326 2   Sprint_fao ('    Dump of fab at !XL:', .fab);
235 0327 2   Sprint_fao ('      15AS   XB      - block id ', %ASCID 'FAB$B_BID', .fab[FAB$B_BID]);
236 0328 2   Sprint_fao ('      15AS   XB      - block len ', %ASCID 'FAB$B_BLN', .fab[FAB$B_BLN]);
237 0329 2   Sprint_fao ('      15AS   XW      - internal file index ', %ASCID 'FAB$WIFI', .fab[FAB$WIFI]);
238 0330 2   Sprint_fao ('      15AS   XL      - file options ', %ASCID 'FAB$L_FOP', .fab[FAB$L_FOP]);
239 0331 2   Sprint_fao ('      15AS   XL      - status ', %ASCID 'FAB$L_STS', .fab[FAB$L_STS]);
240 0332 2   Sprint_fao ('      15AS   XL      - status value ', %ASCID 'FAB$L_STV', .fab[FAB$L_STV]);
241 0333 2   Sprint_fao ('      15AS   XL      - allocation quantity ', %ASCID 'FAB$L_ALQ', .fab[FAB$L_ALQ]);
242 0334 2   Sprint_fao ('      15AS   XW      - default allocation quantity ', %ASCID 'FAB$W_DEQ', .fab[FAB$W_DEQ]);
243 0335 2   Sprint_fao ('      15AS   XB      - file access ', %ASCID 'FAB$B_FAC', .fab[FAB$B_FAC]);
244 0336 2   Sprint_fao ('      15AS   XB      - file sharing ', %ASCID 'FAB$B_SHR', .fab[FAB$B_SHR]);
245 0337 2   Sprint_fao ('      15AS   XL      - user context ', %ASCID 'FAB$L_CTX', .fab[FAB$L_CTX]);
246 0338 2   Sprint_fao ('      15AS   XB      - file organization ', %ASCID 'FAB$B_ORG', .fab[FAB$B_ORG]);
247 0339 2   Sprint_fao ('      15AS   XB      - record format ', %ASCID 'FAB$B_RAT', .fab[FAB$B_RAT]);
248 0340 2   Sprint_fao ('      15AS   XB      - record format ', %ASCID 'FAB$B_RFM', .fab[FAB$B_RFM]);
249 P 0341 2   Sprint_fao ('      15AS   !XB !XL      ''!AF'', %ASCID 'filename',
250 0342 2           .fab[FAB$B_FNS], .fab[FAB$L_FNA], .fab[fab$b_fns], .fab[fab$l_fna]);
251 P 0343 2   Sprint_fao ('      15AS   !XB !XL      ''!AF'', %ASCID 'default name',
252 0344 2           .fab[FAB$B_DNS], .fab[FAB$L_DNA], .fab[fab$b_dns], .fab[fab$l_dna]);
253 0345 2   Sprint_fao ('      15AS   !XL      - device characteristics ', %ASCID 'FAB$L_DEV', .fab[FAB$L_DEV]);
254 0346 2   exch$dbg_utl_print_devchar (.fab[FAB$L_DEV]);
255 0347 2   Sprint_fao ('      15AS   !XL      - nam block address ', %ASCID 'FAB$L_NAM', .fab[FAB$L_NAM]);
256 0348 2   IF .fab[fab$l_nam] NEQ 0
257 0349 2   THEN exch$dbg_nam_dump (.fab[fab$l_nam]);
258 0350 2
259 0351 2 RETURN 1;
260 0352 1 END;

```

.PSECT EXCH\$DBG_PLIT,NOWRT,2

62 61 66 20 66 6F 20 00	70 6D 75 44 20 20 20 74	20 20 20 20 20 20 20 61	20 20 20 20 20 20 20 20	002CC P.AAW:	.ASCII \ Dump of fab at !XL:<0>	:
				010E0017 002DB	.LONG 17694743	:
				00000000 002E4	.ADDRESS P.AAW	:
20 20 53 41 35 31 21 20	20 20 20 57 20 20 20 21	20 20 20 20 20 20 20 21	20 20 20 20 20 20 20 20	002E8 P.AAY:	.ASCII \ !15AS !XW - internal\	:
20 20 20 20 20 20 6C 61	6E 72 65 74 6E 69 66 66	69 65 69 66 69 66 66 66	6C 69 66 66 69 66 66 66	002EC P.AAY:	.ASCII \ !15AS !XW - internal\	:
00 00 20 20 78 65 64 6E	69 20 65 65 69 66 66 66	20 6C 69 66 69 66 66 66	20 6C 69 66 69 66 66 66	0030A P.AAY:	.ASCII \ file index \<0><0><0>	:
				00314 P.AAX:	.LONG 17694773	:
				010E0035 00324	.ADDRESS P.AAY	:
				00000000 00328 P.AAY:	.ASCII \ FAB\$WIFI\<0><0><0>	:
00 00 00 49 46 49 5F 57	24 42 41 46 0032C P.ABA:	20 20 20 20 20 20 20 20	20 20 20 20 20 20 20 20	00338 P.AAZ:	.LONG 17694729	:
69 66 20 20 20 73 6E 6F	69 74 70 6F 20 65 6C 00	20 69 66 66 20 65 6C 00	20 69 66 66 20 65 6C 00	0033C P.AAZ:	.ADDRESS P.ABA	:
				00340 P.ABC:	.ASCII \ !15AS !XL - file options\	:
				0034F P.ABC:	.ASCII \ \<0><0>	:
				0035E P.ABB:	.LONG 17694762	:
				00368 P.ABB:	.ADDRESS P.ABC	:
				010E002A 0036C P.ABB:	.ASCII \ \<0><0>	:
				00000000 00370 P.ABB:	.ADDRESS P.ABC	:

EXCH\$DBG
 V04-000 Conditionally assembled debugging routines N 5
 exch\$dbg_fab_dump 16-Sep-1984 00:45:26 VAX-11 Bliss-32 v4.0-742
 14-Sep-1984 12:29:02 [EXCHNG.SRC]EXCDBG.B32;1 Page 14
 (5) EXC
 V04

20	20	53	41	35	31	21	20	20	20	20	20	20	20	20	00544	P.ACE:	.ASCII \ !15AS !XB - record\	
20	20	20	20	20	20	20	20	42	58	21	20	20	20	20	00553			
				64	72	6F	63	65	72	20	20	2D	20	20	00562			
				00	00	00	20	74	61	6D	72	6F	66	20	0056C	P.ACD:	.ASCII \ format \<0><0><0>	
												010E0031	00578	P.ACG:	.LONG 17694769			
												00000000	0057C	P.AC:	.ADDRESS P.ACE			
												010E0009	00580	P.AC:	.ASCII \FAB\$B RAT\<0><0><0>			
												00000000	0058C	P.AC:	.LONG 17694729			
												00000000	00590	P.AC:	.ADDRESS P.AC			
20	20	53	41	35	31	21	20	20	42	58	21	20	20	20	00594	P.AC:	.ASCII \ !15AS !XB - record\	
20	20	20	20	20	20	20	20	64	72	6F	63	65	72	20	005A3			
								00	00	00	20	74	61	6D	005B2			
													6F	66	20	005BC	.ASCII \ format \<0><0><0>	
													010E0031	005C8	P.AC:	.LONG 17694769		
													00000000	005CC	P.AC:	.ADDRESS P.AC		
													010E0009	005D0	P.AC:	.ASCII \FAB\$B RFM\<0><0><0>		
													00000000	005DC	P.AC:	.LONG 17694729		
													00000000	005E0	P.AC:	.ADDRESS P.AC		
20	20	53	41	35	31	21	20	20	4C	58	20	20	20	20	005E4	P.AC:	.ASCII \ !15AS !XB !XL !!AF"\	
41	21	22	20	20	20	20	20	4C	58	21	20	20	20	20	005F3			
													22	46	00602			
													010E0020	00604	P.AC:	.LONG 17694752		
													00000000	00608	P.AC:	.ADDRESS P.AC		
													65	6D	61	6E	65	6C 69 66 0060C P.AC:
													010E0008	00614	P.AC:	.ASCII \filename\		
													00000000	00618	P.AC:	.LONG 17694728		
													22	46	0063A	P.AC:	.ADDRESS P.AC	
20	21	53	41	35	31	21	20	20	4C	58	21	20	20	20	0061C	P.AC:	.ASCII \ !15AS !XB !XL !!AF"\	
41	21	22	20	20	20	20	20	4C	58	21	20	20	20	20	0062B			
													22	46	0063A			
													010E0020	0063C	P.AC:	.LONG 17694752		
													00000000	00640	P.AC:	.ADDRESS P.AC		
													65	6D	61	6E	65	6C 69 66 00644 P.AC:
													010E000C	00650	P.AC:	.ASCII \default name\		
													00000000	00654	P.AC:	.LONG 17694732		
													22	46	00654	P.AC:	.ADDRESS P.AC	
20	20	53	41	35	31	21	20	20	4C	58	21	20	20	20	00658	P.AC:	.ASCII \ !15AS !XL - device char\	
65	64	20	20	20	20	20	20	61	72	61	68	63	20	65	63	69	76	00667 P.AC:
								20	73	63	69	74	73	69	72	65	74	63 00676 P.AC:
													010E0034	00680	P.AC:	.ASCII \cteristics \		
													00000000	00690	P.AC:	.LONG 17694772		
													00000000	00694	P.AC:	.ADDRESS P.AC		
													010E0009	006A0	P.AC:	.ASCII \FABSL DEV\<0><0><0>		
													00000000	006A4	P.AC:	.LONG 17694729		
													22	46	006A4	P.AC:	.ADDRESS P.AC	
20	20	53	41	35	31	21	20	20	4C	58	21	20	20	20	006A8	P.AC:	.ASCII \ !15AS !XL - nam block ad\	
61	6E	20	20	20	20	20	20	64	61	20	6B	63	6F	6C	62	20	6D	006B7 P.AC:
								00	20	20	73	73	73	65	72	64	006C6 006D0 P.AC:	
													010E002F	006D8	P.AC:	.ASCII \dress \<0>		
													00000000	006DC	P.AC:	.LONG 17694767		
													00000000	006E0	P.AC:	.ADDRESS P.AC		
													010E0009	006EC	P.AC:	.ASCII \FABSL NAM\<0><0><0>		
													00000000	006F0	P.AC:	.LONG 17694729		
													22	46	006F0	P.AC:	.ADDRESS P.AC	

.PSECT EXCH\$DBG_CODE,NOWRT,2

54	00000000G	00	001C	00000	.ENTRY	EXCH\$DBG FAB DUMP, Save R2,R3,R4	: 0321
53	00000000G	EF	9E	00002	MOVAB	LIB\$PUT_OUTPUT, R4	
52	04	AC	DD	00010	MOVAB	EXCH\$UTIL_FAO_BUFFER, R3	: 0326
			52	00014	MOVL	FAB, R2	
	0000'		CF	9F 00016	PUSHL	R2	
63		O2	FB	0001A	PUSHAB	P.AAV	20
			50	DD 0001D	CALLS	#2, EXCH\$UTIL_FAO_BUFFER	41
64		O1	FB	0001F	CALLS	#1, LIB\$PUT_OUTPUT	
7E	02	A2	3C	00022	MOVZWL	2(R2), -(SP)	: 0329
	0000'	CF	9F	00026	PUSHAB	P.AAZ	
	0000'	CF	9F	0002A	PUSHAB	P.AAX	67
63		O3	FB	0002E	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
			50	DD 00031	PUSHL	R0	
64		O1	FB	00033	CALLS	#1, LIB\$PUT_OUTPUT	: 0330
	04	A2	DD	00036	PUSHL	4(R2)	20
	0000'	CF	9F	00039	PUSHAB	P.ABD	41
	0000'	CF	9F	0003D	PUSHAB	P.ABB	
63		O3	FB	00041	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
			50	DD 00044	PUSHL	R0	
64		O1	FB	00046	CALLS	#1, LIB\$PUT_OUTPUT	: 0331
	08	A2	DD	00049	PUSHL	8(R2)	00
	0000'	CF	9F	0004C	PUSHAB	P.ABH	
	0000'	CF	9F	00050	PUSHAB	P.ABF	
63		O3	FB	00054	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	: 20
			50	DD 00057	PUSHL	R0	64
64		O1	FB	00059	CALLS	#1, LIB\$PUT_OUTPUT	
	OC	A2	DD	0005C	PUSHL	12(R2)	: 0332
	0000'	CF	9F	0005F	PUSHAB	P.ABL	
	0000'	CF	9F	00063	PUSHAB	P.ABJ	
63		O3	FB	00067	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
			50	DD 0006A	PUSHL	R0	
64		O1	FB	0006C	CALLS	#1, LIB\$PUT_OUTPUT	: 22
7E	16	A2	9A	0006F	MOVZBL	22(R2), -(SP)	46
	0000'	CF	9F	00073	PUSHAB	P.ABP	
	0000'	CF	9F	00077	PUSHAB	P.ABN	
63		O3	FB	0007B	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
			50	DD 0007E	PUSHL	R0	
64		O1	FB	00080	CALLS	#1, LIB\$PUT_OUTPUT	: 20
7E	17	A2	9A	00083	MOVZBL	23(R2), -(SP)	65
	0000'	CF	9F	00087	PUSHAB	P.ABT	
	0000'	CF	9F	0008B	PUSHAB	P.ABR	72
63		O3	FB	0008F	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
			50	DD 00092	PUSHL	R0	
64		O1	FB	00094	CALLS	#1, LIB\$PUT_OUTPUT	: 0337
	18	A2	DD	00097	PUSHL	24(R2)	
	0000'	CF	9F	0009A	PUSHAB	P.ABX	
	0000'	CF	9F	0009E	PUSHAB	P.ABV	
63		O3	FB	000A2	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
			50	DD 000A5	PUSHL	R0	
64		O1	FB	000A7	CALLS	#1, LIB\$PUT_OUTPUT	
7E	1D	A2	9A	000AA	MOVZBL	29(R2), -(SP)	: 0338
	0000'	CF	9F	000AE	PUSHAB	P.AC	
	0000'	CF	9F	000B2	PUSHAB	P.ABZ	
63		O3	FB	000B6	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
			50	DD 000B9	PUSHL	R0	
64		O1	FB	000BB	CALLS	#1, LIB\$PUT_OUTPUT	

7E	1E	A2	9A	000BE	MOVZBL	30(R2), -(SP)	0339
	0000'	CF	9F	000C2	PUSHAB	P.AC	
	0000'	CF	9F	000C6	PUSHAB	P.AC	
63		03	FB	000CA	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50	DD	000CD	PUSHL	R0	
64		01	FB	000CF	CALLS	#1, LIB\$PUT_OUTPUT	
7E	1F	A2	9A	000D2	MOVZBL	31(R2), -(SP)	0340
	0000'	CF	9F	000D6	PUSHAB	P.ACJ	
	0000'	CF	9F	000DA	PUSHAB	P.AC	
63		03	FB	000DE	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50	DD	000E1	PUSHL	R0	
64		01	FB	000E3	CALLS	#1, LIB\$PUT_OUTPUT	0342
	2C	A2	DD	000E6	PUSHL	44(R2)	
7E	34	A2	9A	000E9	MOVZBL	52(R2), -(SP)	
	2C	A2	DD	000ED	PUSHL	44(R2)	
7E	34	A2	9A	000FO	MOVZBL	52(R2), -(SP)	
	0000'	CF	9F	000F4	PUSHAB	P.ACN	
	0000'	CF	9F	000F8	PUSHAB	P.AC	
63		06	FB	000FC	CALLS	#6, EXCH\$UTIL_FAO_BUFFER	
		50	DD	000FF	PUSHL	R0	
64		01	FB	00101	CALLS	#1, LIB\$PUT_OUTPUT	
	30	A2	DD	00104	PUSHL	48(R2)	0344
7E	35	A2	9A	00107	MOVZBL	53(R2), -(SP)	
	30	A2	DD	0010B	PUSHL	48(R2)	
7E	35	A2	9A	0010E	MOVZBL	53(R2), -(SP)	
	0000'	CF	9F	00112	PUSHAB	P.AC	
	0000'	CF	9F	00116	PUSHAB	P.AC	
63		06	FB	0011A	CALLS	#6, EXCH\$UTIL_FAO_BUFFER	
		50	DD	0011D	PUSHL	R0	
64		01	FB	0011F	CALLS	#1, LIB\$PUT_OUTPUT	
	40	A2	DD	00122	PUSHL	64(R2)	0345
	0000'	CF	9F	00125	PUSHAB	P.AC	
	0000'	CF	9F	00129	PUSHAB	P.AC	
63		03	FB	0012D	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50	DD	00130	PUSHL	R0	
64		01	FB	00132	CALLS	#1, LIB\$PUT_OUTPUT	0346
	40	A2	DD	00135	PUSHL	64(R2)	
0000V	CF	01	FB	00138	CALLS	#1, EXCH\$DBG_UTL_PRINT_DEVCHAR	
	28	A2	DD	0013D	PUSHL	40(R2)	0347
	0000'	CF	9F	00140	PUSHAB	P.AC	
	0000'	CF	9F	00144	PUSHAB	P.ACX	
63		03	FB	00148	CALLS	#3, EXCH\$UTIL_FAO_BUFFER	
		50	DD	0014B	PUSHL	R0	
64		01	FB	0014D	CALLS	#1, LIB\$PUT_OUTPUT	0348
	28	A2	D5	00150	TSTL	40(R2)	
		08	13	00153	BEQL	1S	
	28	A2	DD	00155	PUSHL	40(R2)	0349
0000V	CF	01	FB	00158	CALLS	#1, EXCH\$DBG_NAM_DUMP	
	50	01	DO	0015D	MOVL	#1, R0	0351
		04	00160	1\$:	RET		0352

Routine Size: 353 bytes. Routine Base: EXCH\$DBG_CODE + 0225

```

: 262      0353 1 GLOBAL ROUTINE exch$dbg_fab_name_print (fabb : $ref_bblock) = %SBTTL 'exch$dbg_fab_name_print'
: 263      2 BEGIN
: 264      2
: 265      2 $dbgtrc_prefix ('fab_name_print> ');
: 266      2
: 267      2 LOCAL
: 268      2   tmp_desc : $desc_block,           | A descriptor for the total file name
: 269      2   dev_desc : $desc_block,          | A descriptor for the device name
: 270      2   dir_desc : $desc_block,          | A descriptor for the directory name
: 271      2   dvi_desc : $desc_block,          | A descriptor for the full device name
: 272      2   nam_desc : $desc_block,          | A descriptor for the name name
: 273      2   nod_desc : $desc_block,          | A descriptor for the node name
: 274      2   typ_desc : $desc_block,          | A descriptor for the type name
: 275      2   ver_desc : $desc_block;         | A descriptor for the version number
: 276
: 277      2 BIND
: 278      2   nam = .fabb [fab$l_nam] : $bblock; ! Get pointer to the name block
: 279      2
: 280      2   tmp_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 281      2   dev_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 282      2   dir_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 283      2   dvi_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 284      2   nam_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 285      2   nod_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 286      2   typ_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 287      2   ver_desc [0,0,32,0] = dsc$k_class_s^24 OR dsc$k_dtype_t^16; ! Static desc
: 288
: 289      2 IF .fabb [fab$b_fns] NEQ 0
: 290      2 THEN
: 291      2   BEGIN
: 292      3   tmp_desc [dsc$w_length] = .fabb [fab$b_fns];           | Create file name desc
: 293      3   tmp_desc [dsc$a_pointer] = .fabb [fab$l_fna];          | ...
: 294      3   $print_fao ('FAB filename "!AS"', tmp_desc);
: 295      3   END
: 296      2 ELSE
: 297      2   $print_lit ('FAB filename "" (null)');
: 298
: 299      2 IF NOT (.fabb [fab$l_sts])
: 300      2 THEN
: 301      P 0392 2   $print_fao ('FAB$L_STS = !XL    FAB$L_STV = !XL',
: 302      2   .fabb [fab$l_sts], .fabb [fab$l_stv]);
: 303
: 304      2 IF nam EQL 0
: 305      2 THEN
: 306      2   BEGIN
: 307      3   $print_lit ('No nam block');
: 308      3   RETURN 1;
: 309      2   END;
: 310
: 311      2 IF .nam [nam$b_esl] GTRU 0
: 312      2 THEN
: 313      2   BEGIN
: 314      3   tmp_desc [dsc$w_length] = .nam [nam$b_esl];           | Create file name desc
: 315      3   tmp_desc [dsc$a_pointer] = .nam [nam$l_esl];          | ...
: 316      3   $print_fao ('Expanded string "!AS"', tmp_desc);
: 317
: 318      2 ELSE
: 319      2   BEGIN

```

```

319      0410 2     $print_lit ('Expanded string """ (null)');
320      0411 2
321      0412 2 IF .nam [nam$b_rsl] GTRU 0
322      0413 2 THEN
323      0414 3   BEGIN
324      0415 3     tmp_desc [dsc$w_length] = .nam [nam$b_rsl];      | Create file name desc
325      0416 3     tmp_desc [dsc$sa_pointer] = .nam [nam$l_rsa];    | ...
326      0417 3     $print_fao ('Result string "!"AS'', tmp_desc);
327      0418 2   END;
328
329      0419 2
330      0420 2 dev_desc [dsc$w_length] = .nam [nam$b_dev];      | Create file name desc
331      0421 2 dev_desc [dsc$sa_pointer] = .nam [nam$l_dev];    | ...
332      0422 2 dir_desc [dsc$w_length] = .nam [nam$b_dir];      | Create file name desc
333      0423 2 dir_desc [dsc$sa_pointer] = .nam [nam$l_dir];    | ...
334      0424 2 BEGIN
335      0425 3   BIND
336      0426 3     len = nam [nam$t_dvi] : BYTE;
337      0427 3     dvi_desc [dsc$w_length] = .len;                  ! Create file name desc
338      0428 2 END;
339      0429 2 dvi_desc [dsc$sa_pointer] = nam [nam$t_dvi]+1;    | ...
340      0430 2 nam_desc [dsc$w_length] = .nam [nam$b_name];    | Create file name desc
341      0431 2 nam_desc [dsc$sa_pointer] = .nam [nam$l_name];  | ...
342      0432 2 nod_desc [dsc$w_length] = .nam [nam$b_node];    | Create file name desc
343      0433 2 nod_desc [dsc$sa_pointer] = .nam [nam$l_node];  | ...
344      0434 2 typ_desc [dsc$w_length] = .nam [nam$b_type];    | Create file name desc
345      0435 2 typ_desc [dsc$sa_pointer] = .nam [nam$l_type];  | ...
346      0436 2 ver_desc [dsc$w_length] = .nam [nam$b_ver];    | Create file name desc
347      0437 2 ver_desc [dsc$sa_pointer] = .nam [nam$l_ver];  | ...
348      0438 2
349      P 0439 2 $print_fao ("!"AS" !"AS" !"AS" !"AS" !"AS" !"AS"
350      0440 2           nod_desc, dev_desc, dir_desc, nam_desc, typ_desc, ver_desc, dvi_desc);
351      0441 2
352      0442 2 RETURN 1;
353      0443 1 END;

```

.PSECT EXCH\$DBG_PLIT,NOWRT,2

21 22 20 65 6D 61 6E 65 6C 69 66 00 20 42 41 46 006F4 P.ADC:	.ASCII \FAB filename "!"AS"\<0><0>
010E0012 00703	.LONG 17694738
22 22 20 65 6D 61 6E 65 6C 69 66 00 00 29 6C 6C 75 6E 20 42 41 46 0070C	.ADDRESS P.ADC
010E0016 00710	.ASCII \FAB filename """ (null)\<0><0>
00000000 0071F	.LONG 17694742
4C 58 21 20 3D 20 53 54 53 5F 4C 24 42 41 46 0072C P.ADD:	.ADDRESS P.ADE
00000000 00730	.ASCII \FAB\$L_STS = !XL FAB\$L_STV = !XL\<0>
3D 20 56 54 53 5F 4C 24 42 41 46 20 20 20 20 0074E	.ASCII <0>
00 4C 58 21 20 00753	.LONG 17694754
010E0022 00754	.ADDRESS P.ADF
00000000 00758	.ASCII \No nam block\
6B 63 6F 6C 62 20 6D 61 6E 20 6F 4E 0075C P.ADI:	.LONG 17694732
010E000C 00768 P.ADH:	.ADDRESS P.ADI
00000000 0076C	.ASCII \Expanded string "!"AS"\<0><0><0>
67 6E 69 72 74 73 20 64 65 64 6E 61 70 78 45 00770 P.ADK:	

		00 00 00 22 53 41 21 22 20 0077F	P.ADJ:	.LONG 17694741	:
67	6E 69 72 74 73 20 64 65 64 6E 61 70 78 45 010E0015, 00788	00000000, 0078C	:ADDRESS P.ADK	\Expanded string "" (null)\<0><0><0>	:
20	20 67 6E 69 72 74 73 20 74 6C 75 41 22 53 73 65 52 010E0019, 00790	00000000, 0079F	P.ADM:	.ASCII \Result string "!!AS"\<0><0><0>	:
41	21 22 53 20 22 53 41 21 22 20 22 53 41 21 22 53 41 21 22 010E0015, 007CC	00000000, 007D0	P.ADQ:	.LONG 17694745	:
20	22 53 41 21 22 53 41 21 22 20 22 53 41 21 22 53 41 21 22 00 00 00 22 010E0029, 007D4	00000000, 007E3	P.ADN:	.ADDRESS P.ADO	:
		00 00 00 22 00800	P.ADP:	.ASCII \"!AS\" \"!AS\" \"!AS\" \"!AS\" \"!AS\" \"!AS\" \"!AS\	:
		00000000, 00804	.LONG 17694761	\\""\<0><0><0>	:
			.ADDRESS P.ADQ	:	

.PSECT EXCH\$DBG_CODE,NOWRT,2

		55 00000000G	003C 00000	.ENTRY EXCH\$DBG_FAB_NAME_PRINT, Save R2,R3,R4,R5	0353
		54 00000000G	00 9E 00002	MOVAB LIB\$PUT_OUTPUT, R5	:
		5E	EF 9E 00009	MOVAB EXCH\$UTIL_FAO_BUFFER, R4	:
		52 04	3C C2 00010	SUBL2 #60, SP	0369
		53 28	AC DO 00013	MOVL FABB, R2	0371
34	AE 010E0000	8F	DO 0001B	MOVL #17694720, TMP_DESC	0372
2C	AE 010E0000	8F	DO 00023	MOVL #17694720, DEV_DESC	0373
24	AE 010E0000	8F	DO 0002B	MOVL #17694720, DIR_DESC	0374
1C	AE 010E0000	8F	DO 00033	MOVL #17694720, DVI_DESC	0375
14	AE 010E0000	8F	DO 0003B	MOVL #17694720, NAM_DESC	0376
0C	AE 010E0000	8F	DO 00043	MOVL #17694720, NOD_DESC	0377
04	AE 010E0000	8F	DO 0004B	MCVL #17694720, TYP_DESC	0378
	010E0000	8F	DD 00053	PUSHL #17694720	0380
		34	A2 95 00059	TSTB 52(R2)	0381
			18 13 0005C	BEQL 1\$	0382
38	AE 34	A2 9B 0005E	MOVZBW 52(R2), TMP_DESC	0383	
3C	AE 2C	A2 DO 00063	MOVL 44(R2), TMP_DESC+4	0384	
	38	AE 9F 00068	PUSHAB TMP DESC	0385	
	0000'	CF 9F 0006B	PUSHAB P.ADB		
	64	02 FB 0006F	CALLS #2, EXCH\$UTIL_FAO_BUFFER		
		50 DD 00072	PUSHL R0		
		04 11 00074	BRB 2\$		
	0000'	CF 9F 00076	PUSHAB P.ADD	0388	
	65	01 FB 0007A	CALLS #1, LIB\$PUT_OUTPUT		
10	08 A2	E8 0007D	BLBS 8(R2), 3\$	0390	
7E	08 A2	7D 00081	MOVQ 8(R2), -(SP)	0393	
	0000'	CF 9F 00085	PUSHAB P.ADF		
	64	03 FB 00089	CALLS #3, EXCH\$UTIL_FAO_BUFFER		
		50 DD 0008C	PUSHL R0		
	65	01 FB 0008E	CALLS #1, LIB\$PUT_OUTPUT		
		53 D5 00091	TSTL R3	0395	
	0000'	07 12 00093	BNEQ 4\$		
		CF 9F 00095	PUSHAB P.ADH	0398	
	00A5	31 00099	BRW 8\$		

		0B	A3	95	0009C	4\$:	TSTB	11(R3)	: 0402	
		18	13	0009F			BEQL	5\$		
38	AE	0B	A3	9B	000A1		MOVZBW	11(R3), TMP_DESC	0405	
3C	AE	0C	A3	D0	000A6		MOVL	12(R3), TMP_DESC+4	0406	
		38	AE	9F	000AB		PUSHAB	TMP DESC	0407	
		0000'	CF	9F	000AE		PUSHAB	P.ADJ		
		64	02	FB	000B2		CALLS	#2, EXCH\$UTIL_FAO_BUFFER		
			50	DD	000B5		PUSHL	R0		
			04	11	000B7		BRB	6\$		
		0000'	CF	9F	000B9	5\$:	PUSHAB	P.ADL	0410	
		65	01	FB	000BD	6\$::	CALLS	#1, LIB\$PUT_OUTPUT		
			03	A3	95	000C0	TSTB	3(R3)	0412	
38	AE	03	A3	9B	000C5		BEQL	7\$	0415	
3C	AE	04	A3	D0	000CA		MOVZBW	3(R3), TMP_DESC	0416	
		38	AE	9F	000CF		MOVL	4(R3), TMP_DESC+4	0417	
		0000'	CF	9F	000D2		PUSHAB	TMP DESC		
		64	02	FB	000D6		PUSHAB	P.ADN		
			50	DD	000D9		CALLS	#2, EXCH\$UTIL_FAO_BUFFER		
			01	FB	000DB		PUSHL	R0		
30	AE	39	A3	9B	000DE	7\$::	CALLS	#1, LIB\$PUT_OUTPUT		
34	AE	44	A3	D0	000E3		MOVZBW	57(R3), DEV_DESC	0420	
28	AE	3A	A3	9B	000E8		MOVL	68(R3), DEV_DESC+4	0421	
2C	AE	48	A3	D0	000ED		MOVZBW	58(R3), DIR_DESC	0422	
20	AE	14	A3	9B	000F2		MOVL	72(R3), DIR_DESC+4	0423	
24	AE	15	A3	9E	000F7		MOVAB	20(R3), DVI_DESC	0427	
18	AE	3B	A3	9B	000FC		MOVZBW	21(R3), DVI_DESC+4	0429	
1C	AE	4C	A3	D0	00101		MOVL	59(R3), NAM_DESC	0430	
10	AE	38	A3	9B	00106		MOVZBW	76(R3), NAM_DESC+4	0431	
14	AE	40	A5	D0	0010B		MOVL	56(R3), NOD_DESC	0432	
08	AE	3C	A3	9B	00110		MOVZBW	64(R3), NOD_DESC+4	0433	
0C	AE	50	A3	D0	00115		MOVL	60(R3), TYP_DESC	0434	
04	AE	6E	3D	A3	9B	0011A		MOVZBW	80(R3), TYP_DESC+4	0435
		54	A3	D0	0011E		MOVL	61(R3), VER_DESC	0436	
		20	AE	9F	00123		PUSHAB	84(R3), VER_DESC+4	0437	
		04	AE	9F	00126		PUSHAB	DVI_DESC	0440	
		10	AE	9F	00129		PUSHAB	VER_DESC		
		24	AE	9F	0012C		PUSHAB	TYP_DESC		
		38	AE	9F	0012F		PUSHAB	NAM_DESC		
		44	AE	9F	00132		PUSHAB	DIR_DESC		
		28	AE	9F	00135		PUSHAB	DEV_DESC		
		0000'	CF	9F	00138		PUSHAB	NOD_DESC		
		64	08	FB	0013C		CALLS	#8, EXCH\$UTIL_FAO_BUFFER		
			50	DD	0013F		PUSHL	R0		
		65	01	FB	00141	8\$::	CALLS	#1, LIB\$PUT_OUTPUT		
		50	01	D0	00144		MOVL	#1, R0	0442	
			04	00147			RET		0443	

; Routine Size: 328 bytes. Routine Base: EXCH\$DBG_CODE + 0386

```

354      0444 1 GLOBAL ROUTINE exch$dbg_dump_rt11_file (filb : $ref_bblock) = %SBTTL 'exch$dbg_dump_rt11_file'
355      0445 2 BEGIN
356      0446 2 LOCAL
357      0447 2   buf : VECTOR [512, BYTE],
358      0448 2   pbn,
359      0449 2   cnt,
360      0450 2   status;
361      0451 2
362      0452 2 EXTERNAL ROUTINE
363      0453 2   exch$io_rt11_read;
364      0454 2
365      0455 2 BIND
366      0456 2   bufw = buf : VECTOR [256, WORD],
367      0457 2   ctx = filb [filb$sa_context] : $ref_bblock,
368      0458 2   volb = filb [filb$sa_assoc_volb] : $ref_bblock;
369      0459 2
370      0460 2 $dbgtrc_prefix ('dbg_dump_rt11_file');
371      0461 2
372      0462 2 $block_check (2, .filb, filb, 414);
373      0463 2 $block_check (2, .volb, volb, 415);
374      0464 2 $block_check (2, .ctx, rt11ctx, 416);
375      0465 2 $logic_check (2, (.ctx [rt11ctx$sa_assoc_filb] EQL .filb), 106);
376      0466 2 $logic_check (2, (.ctx [rt11ctx$sa_assoc_volb] EQL .volb), 107);
377      0467 2
378      0468 2
379      0469 2   pbn = .ctx [rt11ctx$1_start_block];
380      0470 2   cnt = .ctx [rt11ctx$w_blocks];
381      0471 2
382      0472 2 INCRU blk FROM .pbn TO .pbn + .cnt - 1
383      0473 2 DO
384      0474 3   BEGIN
385      0475 3
386      0476 4   IF NOT (status = exch$io_rt11_read (.volb, .blk, 1, buf))
387      0477 3   THEN
388      0478 3     Sexch_signal_return (.status);
389      0479 3
390      0480 3     $print_fao ('> !XW Vbn  (!XW Pbn)', .blk-.pbn, .blk);
391      0481 3
392      0482 3     INCRU i FROM 0 TO 15
393      0483 3     DO
394      0484 4       BEGIN
395      0485 4         REGISTER off, offw;
396      0486 4
397      0487 4         off = .i * 32;          offw = .i * 16;
398      0488 4
399      P 0489 4         $print_fao ('> !16(5XW) - !XW - "!AF"',
400      P 0490 4           .bufw [.offw + 15], .bufw [.offw + 14], .bufw [.offw + 13], .bufw [.offw + 12],
401      P 0491 4           .bufw [.offw + 11], .bufw [.offw + 10], .bufw [.offw + 9], .bufw [.offw + 8],
402      P 0492 4           .bufw [.offw + 7], .bufw [.offw + 6], .bufw [.offw + 5], .bufw [.offw + 4],
403      P 0493 4           .bufw [.offw + 3], .bufw [.offw + 2], .bufw [.offw + 1], .bufw [.offw],
404      0494 4           .off, 32, buf [.off]);
405      0495 4
406      0496 3       END;
407      0497 2     END;
408      0498 2
409      0499 2     RETURN 1;
410      0500 1   END;

```

```

.PSECT EXCH$DBG_PLIT,NOWRT,2
58 21 28 20 20 6E 62 56 20 57 58 21 20 3E 00808 P.ADS: .ASCII \> !XW Vbn (!XW Pbn)\<0><0><0>
00 00 00 29 6E 62 50 20 57 00817 010E0015 00820 P.ADR: .LONG 17694741
00000000 00824 .ADDRESS P.ADS
2D 20 29 57 58 35 28 36 31 21 20 20 57 58 20 3E 00828 P.ADU: .ASCII \> !16(5XW) - !XW - ''!AF''\<0>
010E001B 00837 00000000 00844 P.ADT: .LONG 17694747
00000000 00848 .ADDRESS P.ADU

.EXTRN EXCH$IO_RT11_READ
.EXTRN EXCH$_BADLOGIC

.PSECT EXCH$DBG_CODE,NOWRT,2
62
      OFFC 00000
      5B 00000000G EF 9E 00002 .ENTRY EXCH$DBG_DUMP RT11 FILE, Save R2,R3,R4,R5,- : 0444
      5A 00000000G 00 9E 00009 MOVAB EXCH$UTIL_FA0_BUFFER, R11
      59 00000000G 8F D0 00010 MOVAB LIB$STOP,-R10
      58 00000000G EF 9E 00017 MOVL #EXCH$ BADLOGIC, R9
      SE FE00 CE 9E 0001E MOVAB EXCH$UTIL_BLOCK_CHECK, R8
      AC 20 C1 00023 ADDL3 -512(SP), SP
      AC 1C C1 00028 ADDL3 #32, FILB, R3
      52 035B00FA 8F D0 0002D MOVL #56295674, R2
      51 019E 8F 3C 00034 MOVZWL #414, R1
      50 04 AC D0 00039 MOVL FILB, R0
      52 041B00F3 68 16 0003D JSB EXCH$UTIL_BLOCK_CHECK
      51 019F 8F 3C 00046 MOVL #68878579, R2
      50 66 D0 0004B MOVZWL #415, R1
      52 008200F4 68 16 0004E MOVL (R6), R0
      51 01A0 8F 3C 00050 JSB EXCH$UTIL_BLOCK_CHECK
      50 54 D0 0005F MOVL (R3), R4
      04 AC 10 A4 D1 00064 MOVL #8519924, R2
      0B 08 13 00069 MOVZWL #416, R1
      7E 6A 8F 9A 0006B MOVL R4, R0
      01 DD 0006F BEQL EXCH$UTIL_BLOCK_CHECK
      59 DD 00071 PUSHL 1$
      6A 03 FB 00073 1$: CALLS #3, LIB$STOP
      66 14 A4 D1 00076 CMPL 20(R4), (R6)
      0B 08 13 0007A BEQL 2$
      7E 6B 8F 9A 0007C MOVZBL #107, -(SP)
      01 DD 00080 PUSHL R9
      59 DD 00082 PUSHL R9
      6A 03 FB 00084 CALLS #3, LIB$STOP
      53 72 A4 D0 00087 2$: MOVL 114(R4), PBN
      50 40 A4 3C 0008B MOVZWL 64(R4), CNT
      55 FF A043 9E 0008F MOVAB -1(CNT)[PBN], R5
      52 53 D0 00094 MOVL PBN, BLK

```

		00B5	31	00097	BRW	6\$
		5E	DD	0009A	PUSHL	SP
		01	DD	0009C	PUSHL	#1
		52	DD	0009E	PUSHL	BLK
		66	DD	000A0	PUSHL	(R6)
00000000G	EF	04	FB	000A2	CALLS	#4, EXCH\$IO_RT11_READ
57		50	DO	000A9	MOVL	R0, STATUS
10		57	E8	000AC	BLBS	STATUS, 4\$
54		57	DO	000AF	MOVL	STATUS, TEMP
00000000G	00	54	DD	000B2	PUSHL	TEMP
50		01	FB	000B4	CALLS	#1, LIB\$SIGNAL
		54	DO	000BB	MOVL	TEMP, R0
		04	000BE	RET		
7E	52	52	DD	000BF	4\$: PUSHL	BLK
		53	C3	000C1	SUBL3	PBN, BLK, -(SP)
	6B	CF	9F	000C5	PUSHAB	P.ADR
		03	FB	000C9	CALLS	#5, EXCH\$UTIL_FA0_BUFFER
00000000G	00	50	DD	000CC	PUSHL	R0
		01	FB	000CE	CALLS	#1, LIB\$PUT_OUTPUT
50	54	54	D4	000D5	CLRL	I
51	54	05	78	000D7	5\$: ASHL	#5, I, OFF
		04	78	000DB	ASHL	#4, I, OFFW
		6E40	9F	000DF	PUSHAB	BUF[OFF]
		20	DD	000E2	PUSHL	#32
		50	DD	000E4	PUSHL	OFF
7E	0C	AE41	3C	000E6	MOVZWL	BUFW[OFFW], -(SP)
7E	12	AE41	3C	000EB	MOVZWL	BUFW+2[OFFW], -(SP)
7E	18	AE41	3C	000FO	MOVZWL	BUFW+4[OFFW], -(SP)
7E	1E	AE41	3C	000F5	MOVZWL	BUFW+6[OFFW], -(SP)
7E	24	AE41	3C	000FA	MOVZWL	BUFW+8[OFFW], -(SP)
7E	2A	AE41	3C	000FF	MOVZWL	BUFW+10[OFFW], -(SP)
7E	30	AE41	3C	00104	MOVZWL	BUFW+12[OFFW], -(SP)
7E	36	AE41	3C	00109	MOVZWL	BUFW+14[OFFW], -(SP)
7E	3C	AE41	3C	0010E	MOVZWL	BUFW+16[OFFW], -(SP)
7E	42	AE41	3C	00113	MOVZWL	BUFW+18[OFFW], -(SP)
7E	48	AE41	3C	00118	MOVZWL	BUFW+20[OFFW], -(SP)
7E	4E	AE41	3C	0011D	MOVZWL	BUFW+22[OFFW], -(SP)
7E	54	AE41	3C	00122	MOVZWL	BUFW+24[OFFW], -(SP)
7E	5A	AE41	3C	00127	MOVZWL	BUFW+26[OFFW], -(SP)
7E	60	AE41	3C	0012C	MOVZWL	BUFW+28[OFFW], -(SP)
7E	66	AE41	3C	00131	MOVZWL	BUFW+30[OFFW], -(SP)
		0000'	CF	9F	00136	PUSHAB
6B		14	FB	0013A	CALLS	#20, EXCH\$UTIL_FA0_BUFFER
		50	DD	0013D	PUSHL	R0
00000000G	00	01	FB	0013F	CALLS	#1, LIB\$PUT_OUTPUT
		54	D6	00146	INCL	I
OF		54	D1	00148	CMPL	I, #15
		8A	1B	0014B	BLEQU	5\$
		52	D6	0014D	INCL	BLK
55		52	D1	0014F	6\$: CMPL	BLK, R5
		03	1A	00152	BGTRU	7\$
		FF43	31	00154	BRW	3\$
50		01	DO	00157	7\$: MOVL	#1, R0
		04	0015A	RET		

; Routine Size: 347 bytes, Routine Base: EXCH\$DBG_CODE + 04CE

EXCH\$DBG
V04-000

Conditionally assembled debugging routines
exch\$dbg_dump_rt11_file

K 6
16-Sep-1984 00:45:26
14-Sep-1984 12:29:02

VAX-11 Bliss-32 V4.0-742
[EXCHNG.SRC]EXCDBG.B32;1

Page 24
(7)

EXC
V04

69
6E

68
6E

68
69

```

412      0501 1 GLOBAL ROUTINE exch$dbg_nam_dump (nam : $ref_bblock) = %SBTTL 'exch$dbg_nam_dump'
413      0502 2 BEGIN
414      0503 2
415      0504 2 $dbgtrc_prefix ('nam_dump> ');
416      0505 2
417      0506 2
418      0507 2 | A macro to add successive ASCII items to a string buffer, and update the
419      0508 2 | descriptor for the string.
420      0509 2
421      0510 2 MACRO
422      M 0511 2     add_to_list (string) =
423      M 0512 2     BEGIN
424      M 0513 2     BIND desc = %ASCID string : $BBLOCK;
425      M 0514 2     CH$MOVE(.desc[dsc$w_length],
426      M 0515 2             .desc[dsc$w_pointer],
427      M 0516 2             .ptr[1] + .ptr[0]);
428      M 0517 2     ptr[0] = .ptr[0] + .desc[dsc$w_length];
429      M 0518 2     END%;
430      M 0519 2
431      0520 2 LOCAL
432      0521 2     buffer : VECTOR [255, BYTE],
433      0522 2     ptr : VECTOR [2, LONG];
434
435      0524 2     $print_fao ('      Dump of nam at |XL:', .nam);
436      0525 2     | $print_fao ('          |15AS   |XB      - block id ', %ASCID 'NAM$B_BID', .nam[NAM$B_BID]);
437      0526 2     | $print_fao ('          |15AS   |XB      - block length ', %ASCID 'NAM$B_BLN', .nam[NAM$B_BLN]);
438      0527 2     | $print_fao ('          |15AS   |XB      - resultant string area size ', %ASCID 'NAM$B_RSS', .nam[NAM$B_RSS]);
439      0528 2     | $print_fao ('          |15AS   |XB      - name options ', %ASCID 'NAM$B_NOP', .nam[NAM$B_NOP]);
440      0529 2     | $print_fao ('          |15AS   |XL      - wild card context ', %ASCID 'NAM$L_WCC', .nam[NAM$L_WCC]);
441      0530 2     ptr [0] = 0;
442      0531 2     ptr [1] = buffer;
443      0532 2     IF .nam[nam$1_fnb] NEQ 0
444      0533 2     THEN
445      0534 3     BEGIN
446      0535 3     IF .nam[NAM$V_EXP_VER]
447      0536 3     IF .nam[NAM$V_EXP_TYPE]
448      0537 3     IF .nam[NAM$V_EXP_NAME]
449      0538 3     IF .nam[NAM$V_WILD_VER]
450      0539 3     IF .nam[NAM$V_WILD_TYPE]
451      0540 3     IF .nam[NAM$V_WILD_NAME]
452      0541 3     IF .nam[NAM$V_EXP_DIR]
453      0542 3     IF .nam[NAM$V_EXP_DEV]
454      0543 3     IF .nam[NAM$V_WILDCARD]
455      0544 3     IF .nam[NAM$V_SEARCH_LIST]
456      0545 3     IF .nam[NAM$V_CNCL_DEV]
457      0546 3     IF .nam[NAM$V_ROOT_DIR]
458      0547 3     IF .nam[NAM$V_LOWVER]
459      0548 3     IF .nam[NAM$V_HIGHVER]
460      0549 3     IF .nam[NAM$V_PPF]
461      0550 3     IF .nam[NAM$V_NODE]
462      0551 3     IF .nam[NAM$V_QUOTED]
463      0552 3     IF .nam[NAM$V_WILD_DIR]
464      0553 4     IF .nam[NAM$V_GRP_MBR] THEN
465      0554 4
466      0555 4
467      0556 4
468      0557 3
469
470      0558 3     THEN add_to_list('xver ');
471      0559 3     THEN add_to_list('xtyp ');
472      0560 3     THEN add_to_list('xnam ');
473      0561 3     THEN add_to_list('*ver ');
474      0562 3     THEN add_to_list('*typ ');
475      0563 3     THEN add_to_list('*nam ');
476      0564 3     THEN add_to_list('xdir ');
477      0565 3     THEN add_to_list('xdev ');
478      0566 3     THEN add_to_list('***');
479      0567 3     THEN add_to_list('schl ');
480      0568 3     THEN add_to_list('cncl ');
481      0569 3     THEN add_to_list('root ');
482      0570 3     THEN add_to_list('lowv ');
483      0571 3     THEN add_to_list('hiv ');
484      0572 3     THEN add_to_list('ppf ');
485      0573 3     THEN add_to_list('nod ');
486      0574 3     THEN add_to_list('quot ');
487      0575 3     THEN add_to_list('*dir ');
488
489      0576 3     BEGIN
490      0577 3     add_to_list('grpmem ');
491      0578 3     IF .nam[NAM$V_WILD_GRP] THEN add_to_list('*grp ');
492      0579 3     IF .nam[NAM$V_WILD_MBR] THEN add_to_list('*mbr ');
493      0580 3     END;
494
495      0581 3
496      0582 3
497      0583 3
498      0584 3
499      0585 3
500      0586 3
501      0587 3
502      0588 3
503      0589 3
504      0590 3
505      0591 3
506      0592 3
507      0593 3
508      0594 3
509      0595 3
510      0596 3
511      0597 3
512      0598 3
513      0599 3
514      0600 3
515      0601 3
516      0602 3
517      0603 3
518      0604 3
519      0605 3
520      0606 3
521      0607 3
522      0608 3
523      0609 3
524      0610 3
525      0611 3
526      0612 3
527      0613 3
528      0614 3
529      0615 3
530      0616 3
531      0617 3
532      0618 3
533      0619 3
534      0620 3
535      0621 3
536      0622 3
537      0623 3
538      0624 3
539      0625 3
540      0626 3
541      0627 3
542      0628 3
543      0629 3
544      0630 3
545      0631 3
546      0632 3
547      0633 3
548      0634 3
549      0635 3
550      0636 3
551      0637 3
552      0638 3
553      0639 3
554      0640 3
555      0641 3
556      0642 3
557      0643 3
558      0644 3
559      0645 3
560      0646 3
561      0647 3
562      0648 3
563      0649 3
564      0650 3
565      0651 3
566      0652 3
567      0653 3
568      0654 3
569      0655 3
570      0656 3
571      0657 3
572      0658 3
573      0659 3
574      0660 3
575      0661 3
576      0662 3
577      0663 3
578      0664 3
579      0665 3
580      0666 3
581      0667 3
582      0668 3
583      0669 3
584      0670 3
585      0671 3
586      0672 3
587      0673 3
588      0674 3
589      0675 3
590      0676 3
591      0677 3
592      0678 3
593      0679 3
594      0680 3
595      0681 3
596      0682 3
597      0683 3
598      0684 3
599      0685 3
600      0686 3
601      0687 3
602      0688 3
603      0689 3
604      0690 3
605      0691 3
606      0692 3
607      0693 3
608      0694 3
609      0695 3
610      0696 3
611      0697 3
612      0698 3
613      0699 3
614      0700 3
615      0701 3
616      0702 3
617      0703 3
618      0704 3
619      0705 3
620      0706 3
621      0707 3
622      0708 3
623      0709 3
624      0710 3
625      0711 3
626      0712 3
627      0713 3
628      0714 3
629      0715 3
630      0716 3
631      0717 3
632      0718 3
633      0719 3
634      0720 3
635      0721 3
636      0722 3
637      0723 3
638      0724 3
639      0725 3
640      0726 3
641      0727 3
642      0728 3
643      0729 3
644      0730 3
645      0731 3
646      0732 3
647      0733 3
648      0734 3
649      0735 3
650      0736 3
651      0737 3
652      0738 3
653      0739 3
654      0740 3
655      0741 3
656      0742 3
657      0743 3
658      0744 3
659      0745 3
660      0746 3
661      0747 3
662      0748 3
663      0749 3
664      0750 3
665      0751 3
666      0752 3
667      0753 3
668      0754 3
669      0755 3
670      0756 3
671      0757 3
672      0758 3
673      0759 3
674      0760 3
675      0761 3
676      0762 3
677      0763 3
678      0764 3
679      0765 3
680      0766 3
681      0767 3
682      0768 3
683      0769 3
684      0770 3
685      0771 3
686      0772 3
687      0773 3
688      0774 3
689      0775 3
690      0776 3
691      0777 3
692      0778 3
693      0779 3
694      0780 3
695      0781 3
696      0782 3
697      0783 3
698      0784 3
699      0785 3
700      0786 3
701      0787 3
702      0788 3
703      0789 3
704      0790 3
705      0791 3
706      0792 3
707      0793 3
708      0794 3
709      0795 3
710      0796 3
711      0797 3
712      0798 3
713      0799 3
714      0800 3
715      0801 3
716      0802 3
717      0803 3
718      0804 3
719      0805 3
720      0806 3
721      0807 3
722      0808 3
723      0809 3
724      0810 3
725      0811 3
726      0812 3
727      0813 3
728      0814 3
729      0815 3
730      0816 3
731      0817 3
732      0818 3
733      0819 3
734      0820 3
735      0821 3
736      0822 3
737      0823 3
738      0824 3
739      0825 3
740      0826 3
741      0827 3
742      0828 3
743      0829 3
744      0830 3
745      0831 3
746      0832 3
747      0833 3
748      0834 3
749      0835 3
750      0836 3
751      0837 3
752      0838 3
753      0839 3
754      0840 3
755      0841 3
756      0842 3
757      0843 3
758      0844 3
759      0845 3
760      0846 3
761      0847 3
762      0848 3
763      0849 3
764      0850 3
765      0851 3
766      0852 3
767      0853 3
768      0854 3
769      0855 3
770      0856 3
771      0857 3
772      0858 3
773      0859 3
774      0860 3
775      0861 3
776      0862 3
777      0863 3
778      0864 3
779      0865 3
780      0866 3
781      0867 3
782      0868 3
783      0869 3
784      0870 3
785      0871 3
786      0872 3
787      0873 3
788      0874 3
789      0875 3
790      0876 3
791      0877 3
792      0878 3
793      0879 3
794      0880 3
795      0881 3
796      0882 3
797      0883 3
798      0884 3
799      0885 3
800      0886 3
801      0887 3
802      0888 3
803      0889 3
804      0890 3
805      0891 3
806      0892 3
807      0893 3
808      0894 3
809      0895 3
810      0896 3
811      0897 3
812      0898 3
813      0899 3
814      0900 3
815      0901 3
816      0902 3
817      0903 3
818      0904 3
819      0905 3
820      0906 3
821      0907 3
822      0908 3
823      0909 3
824      0910 3
825      0911 3
826      0912 3
827      0913 3
828      0914 3
829      0915 3
830      0916 3
831      0917 3
832      0918 3
833      0919 3
834      0920 3
835      0921 3
836      0922 3
837      0923 3
838      0924 3
839      0925 3
840      0926 3
841      0927 3
842      0928 3
843      0929 3
844      0930 3
845      0931 3
846      0932 3
847      0933 3
848      0934 3
849      0935 3
850      0936 3
851      0937 3
852      0938 3
853      0939 3
854      0940 3
855      0941 3
856      0942 3
857      0943 3
858      0944 3
859      0945 3
860      0946 3
861      0947 3
862      0948 3
863      0949 3
864      0950 3
865      0951 3
866      0952 3
867      0953 3
868      0954 3
869      0955 3
870      0956 3
871      0957 3
872      0958 3
873      0959 3
874      0960 3
875      0961 3
876      0962 3
877      0963 3
878      0964 3
879      0965 3
880      0966 3
881      0967 3
882      0968 3
883      0969 3
884      0970 3
885      0971 3
886      0972 3
887      0973 3
888      0974 3
889      0975 3
890      0976 3
891      0977 3
892      0978 3
893      0979 3
894      0980 3
895      0981 3
896      0982 3
897      0983 3
898      0984 3
899      0985 3
900      0986 3
901      0987 3
902      0988 3
903      0989 3
904      0990 3
905      0991 3
906      0992 3
907      0993 3
908      0994 3
909      0995 3
910      0996 3
911      0997 3
912      0998 3
913      0999 3
914      0999 3
915      0999 3
916      0999 3
917      0999 3
918      0999 3
919      0999 3
920      0999 3
921      0999 3
922      0999 3
923      0999 3
924      0999 3
925      0999 3
926      0999 3
927      0999 3
928      0999 3
929      0999 3
930      0999 3
931      0999 3
932      0999 3
933      0999 3
934      0999 3
935      0999 3
936      0999 3
937      0999 3
938      0999 3
939      0999 3
940      0999 3
941      0999 3
942      0999 3
943      0999 3
944      0999 3
945      0999 3
946      0999 3
947      0999 3
948      0999 3
949      0999 3
950      0999 3
951      0999 3
952      0999 3
953      0999 3
954      0999 3
955      0999 3
956      0999 3
957      0999 3
958      0999 3
959      0999 3
960      0999 3
961      0999 3
962      0999 3
963      0999 3
964      0999 3
965      0999 3
966      0999 3
967      0999 3
968      0999 3
969      099
```

```

469      0558 3      IF .ptr[0] GTR 0 THEN ptr[0] = .ptr[0] - 1;      ! Remove final space
470      0559 2      END;
471      0560 2      $print_fao ('          !15AS    !XL    - name bits "'AS'", %ACID 'NAMSL_FNB', .nam[NAMSL_FNB], ptr);
472      P 0561 2      $print_fao ('          !15AS    !XB !XL    -!AF", %ACID 'expanded string',
473      P 0562 2      $print_fao ('          !15AS    !XB !XL    .nam[NAMSB_ESL], nam[NAMSL_ESA], .nam[nam$b_esl], .nam[nam$l_esa]);
474      P 0563 2      $print_fao ('          !15AS    !XB !XL    -!AF", %ACID 'result string',
475      P 0564 2      $print_fao ('          !15AS    !AC    .nam[NAMSB_RSL], nam[NAMSL_RSA], .nam[nam$b_rsl], .nam[nam$l_rsa]);
476      P 0565 2      $print_fao ('          !15AS    - device name; %ACID 'NAMST_DVI', nam[NAMST_DVI]);
477      P 0566 2      $print_fao ('          '!AF" '!AF" '!AF" '!AF" '!AF"
478      P 0567 2      .nam[NAMSB_NODE], .nam[NAMSL_NODE],
479      P 0568 2      .nam[NAMSB_DEV], .nam[NAMSL_DEV],
480      P 0569 2      .nam[NAMSB_DIR], .nam[NAMSL_DIR],
481      P 0570 2      .nam[NAMSB_NAME], .nam[NAMSL_NAME],
482      P 0571 2      .nam[NAMSB_TYPE], .nam[NAMSL_TYPE],
483      P 0572 2      .nam[NAMSB_VER], .nam[NAMSL_VER]);
484      0573 2      $print_fao ('          !15AS    !XL    - related file nam block addr ', %ACID 'NAMSL_RLF', .nam[NAMSL_
485      0574 2      IF .nam[NAMSL_RLF] NEQ 0
486      0575 2      THEN exch$dbg_nam_dump (.nam[NAMSL_RLF]);
487      0576 2
488      0577 2      RETURN 1;
489      0578 1      END;

```

.PSECT EXCH\$DBG_PLIT,NOWRT,2

6D	61	6E	20	66	6F	20	00	70	3A	6D	75	44	20	20	20	74	20	20	0084C	P.ADW:	.ASCII \ Dump of nam at !XL:\<0>
20	20	53	41	35	31	21	20	20	42	58	21	20	20	20	20	20	20	20	0085B	P.ADV:	.LONG 17694743
20	20	20	20	20	20	6F	20	65	6D	61	6E	20	20	20	2D	20	20	00868	.ADDRESS P.ADW		
20	20	53	41	35	31	21	20	00	20	73	6E	20	6F	69	74	70	0086C	P.ADY:	.ASCII \ 15AS !XB - name o\		
20	20	20	20	20	20	6F	20	65	6D	61	6E	20	20	20	2D	20	20	0087B			
20	20	20	20	20	20	6F	20	65	6D	61	6E	20	20	20	2D	20	20	0088A			
20	20	20	20	20	20	6F	20	65	6D	61	6E	20	20	20	2D	20	20	00894	.ASCII \ptions \<0>		
20	20	20	20	20	20	6F	20	65	6D	61	6E	20	20	20	2D	20	20	0089C	PADX:	.LONG 17694767	
20	20	20	20	20	20	6F	20	65	6D	61	6E	20	20	20	2D	20	20	008A0	.ADDRESS P.ADY		
20	20	20	20	20	20	6F	20	65	6D	61	6E	20	20	20	2D	20	20	008A4	P.AEA:	.ASCII \NAMS B NOP\<0><0><0>	
20	20	20	20	20	20	6F	20	65	6D	61	6E	20	20	20	2D	20	20	008B0	P.ADZ:	.LONG 17694729	
20	20	20	20	20	20	6F	20	65	6D	61	6E	20	20	20	2D	20	20	008B4	.ADDRESS P.AEA		
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	008B8	P.AEC:	.ASCII \ 15AS !XL - wild card co\		
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	008C7				
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	008D6				
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	008E0	.ASCII \ntext \<0>			
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	008E8	P.AEB:	.LONG 17694767		
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	008EC	.ADDRESS P.AEC			
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	008F0	P.AEE:	.ASCII \NAMS L WCC\<0><0><0>		
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	008FC	P.AED:	.LONG 17694729		
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	00900	.ADDRESS P.AEE			
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	00904	P.AEG:	.ASCII \xver \<0><0><0>		
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	0090C	P.AEF:	.LONG 17694725		
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	00910	.ADDRESS P.AEG			
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	00914	P.AEI:	.ASCII \xtyp \<0><0><0>		
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	0091C	P.AEH:	.LONG 17694725		
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	00920	.ADDRESS P.AEI			
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	00924	P.AEK:	.ASCII \xnam \<0><0><0>		
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	0092C	P.AEJ:	.LONG 17694725		
20	20	20	20	20	20	6F	20	63	20	64	72	61	63	20	64	6C	00930	.ADDRESS P.AEK			

20 20 53 41 35 31 21 20 20 20 20 20 20 20 20 20 20 20 20 00A40 P.AFW: .ASCII \ !15AS !XL - name bits "!"\
61 6E 20 2D 20 20 21 22 20 73 74 69 62 20 65 6D 00A44 00A53 00A62 .ADDRESS P.AFU

		00 22 53 41 00A6C .ASCII \AS'\<0>
		010E002B 00A70 P.AFV: .LONG 17694763
		00000000' 00A74 .ADDRESS P.AFW
	00 00 00 42 4E 46 5F 4C 24 4D 41 4E 00A78 P.AFY: .ASCII \NAM\$L_FNB\<0><0><0>	010E0009 00A84 PAFX: .LONG 17694729
		00000000' 00A88 .ADDRESS P.AFY
20 21 53 41 35 31 21 20 20 4C 20 58 21 20 20 20 20 20 00A8C P.AGA: .ASCII \ !15AS !XB !XL !!AF'\	21 20 20 00A9B 00AAA .	
		22 46 00AAC P.AFZ: .LONG 17694752
67 6E 69 72 74 73 20 64 65 64 6E 61 70 78 65 00AB0 P.AGA: .ADDRESS P.AGA	00000000' 00AB4 P.AGC: .ASCII \expanded string\<0>	
		00 00 00 00 00AC3 00AC4 P.AGB: .LONG 17694735
20 21 53 41 35 31 21 20 20 4C 20 58 21 20 20 20 20 20 00ACC P.AGE: .ASCII \ !15AS !XB !XL !!AF'\	00000000' 00AC8 P.AGC: .ADDRESS P.AGC	
		22 46 00AE4 P.AGD: .LONG 17694752
00 00 67 6E 69 72 74 73 20 74 6C 75 73 65 72 00AF0 P.AGE: .ADDRESS P.AGE	00000000' 00AF4 P.AGG: .ASCII \result string\<0><0><0>	
		00 00 00 00 00B03 00B04 P.AGF: .LONG 17694733
20 20 53 41 35 31 21 20 20 4C 20 58 21 20 20 20 20 20 00B0C P.AGI: .ASCII \ !15AS !!AC" - device name\	00000000' 00B08 P.AGG: .ADDRESS P.AGG	
		22 46 65 6D 61 6E 20 65 63 69 76 65 00B2A 010E0028 00B34 P.AGH: .LONG 17694760
		00000000' 00B38 P.AGI: .ADDRESS P.AGI
	00 00 00 49 56 44 5F 54 24 4D 41 4E 00B3C P.AGK: .ASCII \NAM\$T_DVI\<0><0><0>	010E0009 00B48 P.AGJ: .LONG 17694729
		00000000' 00B4C P.AGM: .ADDRESS P.AGK
22 20 22 46 41 21 22 20 20 20 20 20 20 20 20 20 00B50 P.AGM: .ASCII \ !!AF" !!AF" !!AF" !!AF" !!AF" !!\	21 20 20 22 20 22 00 22 46 41 00B5F 010E002B 00B78 P.AGL: .ASCII \AF'\<0>	
		21 22 20 22 46 41 21 22 20 20 20 20 20 20 00B6E 00B7C P.AGL: .LONG 17694763
20 20 53 41 35 31 21 20 20 4C 20 58 21 20 20 20 20 20 00B84 P.AGO: .ADDRESS P.AGM	00000000' 00B80 P.AGO: .ASCII \ !15AS !XL - related file\	
65 72 20 2D 20 20 65 6C 69 66 20 64 65 74 61 6C 00BA2 010E0039 00BC0 P.AGN: .LONG 17694777	00000000' 00B93 P.AGO: .ADDRESS P.AGO	
72 64 64 61 20 6B 63 6F 6C 62 20 6D 61 6E 20 00BAC 00BC4 P.AGN: .ASCII \ nam block addr \<0><0><0>	00000000' 00BD4 P.AGP: .LONG 17694729	
		00 00 00 46 4C 52 5F 4C 24 4D 41 4E 00BC8 P.AGQ: .ASCII \NAM\$L_RLF\<0><0><0>
		00000000' 00BD8 P.AGP: .ADDRESS P.AGQ

DESC=	P.AEF
DESC=	P.AEH
DESC=	P.AEJ
DESC=	P.AEL
DESC=	P.AEN
DESC=	P.AEP
DESC=	P.AER
DESC=	P.AET

DESC= P.AEV
 DESC= P.AEX
 DESC= P.AEZ
 DESC= P.AFB
 DESC= P.AFD
 DESC= P.AFF
 DESC= P.AFH
 DESC= P.AFJ
 DESC= P.AFL
 DESC= P.AFN
 DESC= P.AFP
 DESC= P.AFR
 DESC= P.AFT

.PSECT EXCH\$DBG_CODE,NOWRT,2

		07FC 00000	.ENTRY	EXCH\$DBG_NAM_DUMP, Save R2,R3,R4,R5,R6,R7,- ; 0501
5A	00000000G	00 9E 00002	MOVAB	LIB\$PUT_OUTPUT, R10
59	00000000G	EF 9E 00009	MOVAB	EXCH\$UTIL_FAO_BUFFER, R9
58	0000'	CF 9E 00010	MOVAB	DESC, R8
5E	FEF8	CE 9E 00015	MOVAB	-264(SP), SP
56	04	AC D0 0001A	MOVL	NAM, R6
		56 DD 0001E	PUSHL	R6
	FF58	C8 9F 00020	PUSHAB	P.ADV
69		02 FB 00024	CALLS	#2, EXCH\$UTIL_FAO_BUFFER
		50 DD 00027	PUSHL	R0
6A		01 FB 00029	CALLS	#1, LIB\$PUT_OUTPUT
7E	08	A6 9A 0002C	MOVZBL	8(R6), -(SP)
	A4	A8 9F 00030	PUSHAB	P.ADZ
	90	A8 9F 00033	PUSHAB	P.ADX
69		03 FB 00036	CALLS	#3, EXCH\$UTIL_FAO_BUFFER
		50 DD 00039	PUSHL	R0
6A		01 FB 0003B	CALLS	#1, LIB\$PUT_OUTPUT
	30	A6 DD 0003E	PUSHL	48(R6)
F0		A8 9F 00041	PUSHAB	P.AED
	DC	A8 9F 00044	PUSHAB	P.AEB
69		03 FB 00047	CALLS	#3, EXCH\$UTIL_FAO_BUFFER
		50 DD 0004A	PUSHL	R0
6A		01 FB 0004C	CALLS	#1, LIB\$PUT_OUTPUT
	6E	D4 0004F	CLRL	PTR
04	AE	08 AE 00051	MOVAB	BUFFER, PTR+4
	57	34 A6 9E 00056	MOVAB	52(R6), R7
		67 D5 0005A	TSTL	(R7)
		03 12 0005C	BNEQ	1\$
		01F6 31 0005E	BRW	22\$
50	04	10 AE 00061	1\$: BLBC	(R7), 2\$
60	04	AE 6E C1 00064	ADDL3	PTR, PTR+4, R0
	B8	68 28 00069	MOVC3	DESC, @DESC+4, (R0)
	50	68 3C 0006E	MOVZWL	DESC, R0
12	6E	50 C0 00071	ADDL2	R0, PTR
50	67	01 E1 00074	BBC	#1, (R7), 3\$
60	04	AE 6E C1 00078	ADDL3	PTR, PTR+4, R0
	14	B8 28 0007D	MOVC3	DESC, @DESC+4, (R0)
	50	10 A8 3C 00083	MOVZWL	DESC, R0
	6E	50 C0 00087	ADDL2	R0, PTR

12		67		02	E1	0008A	3\$:	BBC	#2, (R7), 4\$		0537	
50	04	AE		6E	C1	0008E		ADDL3	PTR, PTR+4, R0			
60	24	B8	50	20	A8	28	00093	MOVC3	DESC, @DESC+4, (R0)			
			6E	20	A8	3C	00099	MOVZWL	DESC, R0			
12		67		03	E1	000A0	4\$:	ADDL2	R0, PTR		0538	
50	04	AE		6E	C1	000A4		BBC	#3, (R7), 5\$			
60	34	B8	50	30	A8	28	000A9	ADDL3	PTR, PTR+4, R0			
			6E	30	A8	3C	000AF	MOVC3	DESC, @DESC+4, (R0)			
12		67		50	C0	000B3		MOVZWL	DESC, R0		0539	
50	04	AE		04	E1	000B6	5\$::	ADDL2	R0, PTR			
60	44	B8	50	40	6E	C1	000BA	BBC	#4, (R7), 6\$			
			6E	40	A8	28	000BF	ADDL3	PTR, PTR+4, R0			
12		67		50	A8	3C	000C5	MOVC3	DESC, @DESC+4, (R0)			
50	04	AE		50	C0	000C9		MOVZWL	DESC, R0			
60	54	B8	50	50	05	E1	000CC	ADDL2	R0, PTR		0540	
			6E	50	6E	C1	000D0	BBC	#5, (R7), 7\$			
12		67		50	A8	28	000D5	ADDL3	PTR, PTR+4, R0			
50	04	AE		50	A8	3C	000DB	MOVC3	DESC, @DESC+4, (R0)			
60	64	B8	50	60	50	C0	000DF	MOVZWL	DESC, R0			
			6E	60	06	E1	000E2	ADDL2	R0, PTR		0541	
12		67		60	6E	C1	000E6	BBC	#6, (R7), 8\$			
50	04	AE		50	A8	28	000EB	ADDL3	PTR, PTR+4, R0			
60	64	B8	50	60	A8	3C	000F1	MOVC3	DESC, @DESC+4, (R0)			
			6E	60	50	C0	000F5	MOVZWL	DESC, R0			
12		67		60	67	95	000F8	ADDL2	R0, PTR		0542	
50	04	AE		60	12	18	000FA	TSTB	(R7)			
60	74	B8	50	70	6E	C1	000FC	BGEQ	9\$			
			6E	70	A8	28	00101	ADDL3	PTR, PTR+4, R0			
15		13		70	A8	3C	00107	MOVC3	DESC, @DESC+4, (R0)		0543	
50	04	AE		50	70	C0	0010B	MOVZWL	DESC, R0			
60	0080	D8	6E	01	A7	E9	0010E	ADDL2	R0, PTR			
			13	7C	6E	C1	00112	BLBC	1(R7), 10\$			
15		67		7C	A8	28	00117	ADDL3	PTR, PTR+4, R0			
50	04	AE		7C	A8	3C	0011E	MOVC3	DESC, @DESC+4, (R0)			
60	0090	D8	6E	50	50	C0	00122	MOVZWL	DESC, R0			
			08C	008C	0B	E1	00125	ADDL2	R0, PTR		0544	
15		67		008C	6E	C1	00129	BBC	#11, (R7), 11\$			
50	04	AE		008C	C8	28	0012E	ADDL3	PTR, PTR+4, R0			
60	0090	D8	50	008C	C8	3C	00136	MOVC3	DESC, @DESC+4, (R0)			
			6E	50	50	C0	0013B	MOVZWL	DESC, R0			
15		67		50	0C	E1	0013E	ADDL2	R0, PTR		0545	
50	04	AE		50	6E	C1	00142	BBC	#12, (R7), 12\$			
60	00A0	D8	6E	009C	C8	28	00147	ADDL3	PTR, PTR+4, R0			
			009C	50	C8	3C	0014F	MOVC3	DESC, @DESC+4, (R0)			
15		67		50	50	C0	00154	MOVZWL	DESC, R0			
50	04	AE		6E	0D	E1	00157	ADDL2	R0, PTR		0546	
60	00B0	D8	50	00AC	6E	C1	0015B	BBC	#13, (R7), 13\$			
			00AC	50	C8	28	00160	ADDL3	PTR, PTR+4, R0			
15		67		50	50	C8	3C	00168	MOVC3	DESC, @DESC+4, (R0)		
50	04	AE		6E	50	C0	0016D	MOVZWL	DESC, R0			
60	00C0	D8	50	00BC	0E	E1	00170	ADDL2	R0, PTR		0547	
			00BC	50	6E	C1	00174	BBC	#14, (R7), 14\$			
15		67		50	50	C8	28	00179	ADDL3	PTR, PTR+4, R0		
50	04	AE		6E	50	C8	3C	00181	MOVC3	DESC, @DESC+4, (R0)		
60	00C0	D8	6E	00BC	50	C0	00186	MOVZWL	DESC, R0			
			67	67	B5	00189	67: 14\$:	ADDL2	R0, PTR		0548	
								TSTW	(R7)			

50	04	AE		15	18	0018B	BGEQ	15\$		
60	00CC	D8	00C8	6E	C1	0018D	ADDL3	PTR	PTR+4, R0	
		50	00C8	C8	28	00192	MOVC3	DESC,	@DESC+4, (R0)	
		6E		C8	3C	0019A	MOVZWL	DESC,	RO	
		15	02	50	C0	0019F	ADDL2	RO, PTR		
50	04	AE		A7	E9	001A2	BLBC	2(R7), 16\$		
60	00D8	D8	00D4	6E	C1	001A6	ADDL3	PTR	PTR+4, R0	
		50	00D4	C8	28	001AB	MOVC3	DESC,	@DESC+4, (R0)	
		6E		C8	3C	001B3	MOVZWL	DESC,	RO	
15	04	AE		50	C0	001B8	ADDL2	RO, PTR		
50	00E4	D8	00E0	11	E1	001BB	BBC	#17, (R7), 17\$		
60		50	00E0	6E	C1	001BF	ADDL3	PTR	PTR+4, R0	
		6E		C8	28	001C4	MOVC3	DESC,	@DESC+4, (R0)	
		15		C8	3C	001CC	MOVZWL	DESC,	RO	
50	04	AE		50	C0	001D1	ADDL2	RO, PTR		
60	00F4	D8	00F0	12	E1	001D4	BBC	#18, (R7), 18\$		
		50	00F0	6E	C1	001D8	ADDL3	PTR	PTR+4, R0	
		6E		C8	28	001DD	MOVC3	DESC,	@DESC+4, (R0)	
		15		C8	3C	001E5	MOVZWL	DESC,	RO	
50	04	AE		50	C0	001EA	ADDL2	RO, PTR		
60	0104	D8	0100	14	E1	001ED	BBC	#20, (R7), 19\$		
		50	0100	6E	C1	001F1	ADDL3	PTR	PTR+4, R0	
		6E		C8	28	001F6	MOVC3	DESC,	@DESC+4, (R0)	
		47		C8	3C	001FE	MOVZWL	DESC,	RO	
50	04	AE		50	C0	00203	ADDL2	RO, PTR		
60	0114	D8	0110	13	E1	00206	BBC	#19, (R7), 21\$		
		50	0110	6E	C1	0020A	ADDL3	PTR	PTR+4, R0	
		6E		C8	28	0020F	MOVC3	DESC,	@DESC+4, (R0)	
		50		C8	3C	00217	MOVZWL	DESC,	RO	
50	04	AE		50	C0	0021C	ADDL2	RO, PTR		
60	0124	D8	0120	15	A7	E9	BLBC	3(R7), 20\$		
		50	0120	03	E6	C1	00223	ADDL3	PTR	PTR+4, R0
		6E		C8	28	00228	MOVC3	DESC,	@DESC+4, (R0)	
		50		C8	3C	00230	MOVZWL	DESC,	RO	
		15		50	C0	00235	ADDL2	RO, PTR		
50	04	AE		67	E1	00238	BBC	#25, (R7), 21\$		
60	0134	D8	0130	6E	C1	0023C	ADDL3	PTR	PTR+4, R0	
		50	0130	19	E1	00241	MOVC3	DESC,	@DESC+4, (R0)	
		6E		C8	28	00249	MOVZWL	DESC,	RO	
		15		50	C0	0024E	ADDL2	RO, PTR		
50	04	AE		67	D5	00251	TSTL	PTR		
60	0134	D8	0130	02	15	00253	BLEQ	22\$		
		6E		6E	D7	00255	DECL	PTR		
		15		50	DD	00257	PUSHL	SP		
		6E		67	DD	00259	PUSHL	(R7)		
		69		0178	C8	9F	PUSHAB	P.AFX		
		0164		0164	C8	9F	PUSHAB	P.AFV		
		69		04	FB	00263	CALLS	#4, EXCH\$UTIL_FAO_BUFFER		
		6A		50	DD	00266	PUSHL	RO		
		6A		01	FB	00268	CALLS	#1, LIB\$PUT_OUTPUT		
		7E		08	A6	DD	PUSHL	12(R6)		
		OC		A6	9A	0026E	MOVZBL	11(R6), -(SP)		
		7E		OC	A6	DD	PUSHL	12(R6)		
		7E		OB	A6	9A	MOVZBL	11(R6), -(SP)		
		01B8		01AO	C8	9F	PUSHAB	P.AGB		
		69		01AO	C8	9F	PUSHAB	P.AFZ		
		69		06	FB	00281	CALLS	#6, EXCH\$UTIL_FAO_BUFFER		

		50	DD	00284	PUSHL	R0		;
		01	FB	00286	CALLS	#1, LIB\$PUT_OUTPUT		1
6A		04	A6	DD 00289	PUSHL	4(R6)		1
7E		03	A6	9A 0028C	MOVZBL	3(R6), -(SP)		1
		04	A6	DD 00290	PUSHL	4(R6)		1
7E		03	A6	9A 00293	MOVZBL	3(R6), -(SP)		1
	01F8	C8	9F	00297	PUSHAB	P.AGF		1
	01E0	C8	9F	0029B	PUSHAB	P.AGD		1
69		06	FB	0029F	CALLS	#6, EXCH\$UTIL_FAO_BUFFER		1
		50	DD	002A2	PUSHL	R0		1
6A		01	FB	002A4	CALLS	#1, LIB\$PUT_OUTPUT		1
	14	A6	9F	002A7	PUSHAB	20(R6)		1
023C		C8	9F	002AA	PUSHAB	P.AGJ		1
0228		C8	9F	002AE	PUSHAB	P.AGH		1
69		03	FB	002B2	CALLS	#3, EXCH\$UTIL_FAO_BUFFER		1
		50	DD	002B5	PUSHL	R0		1
6A		01	FB	002B7	CALLS	#1, LIB\$PUT_OUTPUT		1
7E		54	A6	DD 002BA	PUSHL	84(R6)		1
	3D	A6	9A	002BD	MOVZBL	61(R6), -(SP)		1
7E		50	A6	DD 002C1	PUSHL	80(R6)		1
7E		3C	A6	9A 002C4	MOVZBL	60(R6), -(SP)		1
	4C	A6	DD	002C8	PUSHL	76(R6)		1
7E		3B	A6	9A 002CB	MOVZBL	59(R6), -(SP)		1
	48	A6	DD	002CF	PUSHL	72(R6)		1
7E		3A	A6	9A 002D2	MOVZBL	58(R6), -(SP)		1
	44	A6	DD	002D6	PUSHL	68(R6)		1
7E		39	A6	9A 002D9	MOVZBL	57(R6), -(SP)		1
	40	A6	DD	002DD	PUSHL	64(R6)		1
7E		38	A6	9A 002E0	MOVZBL	56(R6), -(SP)		1
	0270	C8	9F	002E4	PUSHAB	P.AGL		1
69		0D	FB	002E8	CALLS	#13, EXCH\$UTIL_FAO_BUFFER		1
		50	DD	002EB	PUSHL	R0		1
6A		01	FB	002ED	CALLS	#1, LIB\$PUT_OUTPUT		1
	10	A6	DD	002F0	PUSHL	16(R6)		1
02C8		C8	9F	002F3	PUSHAB	P.AGP		1
02B4		C8	9F	002F7	PUSHAB	P.AGN		1
69		03	FB	002FB	CALLS	#3, EXCH\$UTIL_FAO_BUFFER		1
		50	DD	002FE	PUSHL	R0		1
6A		01	FB	00300	CALLS	#1, LIB\$PUT_OUTPUT		1
	10	A6	D5	00303	TSTL	16(R6)		1
	08	13	00306		BEQL	23\$		1
	10	A6	DD	00308	PUSHL	16(R6)		1
FCFO	CF	01	FB	0030B	CALLS	#1, EXCH\$DBG_NAM_DUMP		1
	50	01	DO	00310	23\$:	MOVL	#1, R0	0577
		04	00313			RET		0578

: Routine Size: 788 bytes, Routine Base: EXCH\$DBG_CODE + 0629

```
: 491      0579 1 GLOBAL ROUTINE exch$dbg_utl_print_devchar (dev : $bblock) = %SBTTL 'exch$dbg_utl_print_devchar'
492      0580 2 BEGIN
493
494      0582 2     $dbgtrc_prefix ('dbg_utl_print_devchar ');
495      0583 2     $trace_print_fao ('characteristics longword !XL', .dev [0,0,32,0]);
496
497      0585 2     IF .dev [dev$v_rec]
498      0586 2       $print_lit (' rec - device is record oriented');
499      0587 2     IF .dev [dev$v_ccl]
500      0588 2       $print_lit (' ccl - carriage control device');
501      0589 2     IF .dev [dev$v_trm]
502      0590 2       $print_lit (' trm - device is a terminal');
503      0591 2     IF .dev [dev$v_dir]
504      0592 2       $print_lit (' dir - device is directory structured');
505      0593 2     IF .dev [dev$v_sdi]
506      0594 2       $print_lit (' sdi - device is single directory structured');
507      0595 2     IF .dev [dev$v_sqd]
508      0596 2       $print_lit (' sqd - sequential block-oriented device (i.e., magtape)');
509      0597 2     IF .dev [dev$v_spl]
510      0598 2       $print_lit (' spl - device being spooled');
511      0599 2     IF .dev [dev$v_opr]
512      0600 2       $print_lit (' opr - device is an operator');
513      0601 2     IF .dev [dev$v_rct]
514      0602 2       $print_lit (' rct - disk contains rct (dec standard 166 disk)');
515      0603 2     IF .dev [dev$v_net]
516      0604 2       $print_lit (' net - network device');
517      0605 2     IF .dev [dev$v_fod]
518      0606 2       $print_lit (' fod - files-oriented device (i.e., disk and mt)');
519      0607 2     IF .dev [dev$v_dua]
520      0608 2       $print_lit (' dua - device is dual ported');
521      0609 2     IF .dev [dev$v_shr]
522      0610 2       $print_lit (' shr - device shareable');
523      0611 2     IF .dev [dev$v_gen]
524      0612 2       $print_lit (' gen - device is a generic device');
525      0613 2     IF .dev [dev$v_avl]
526      0614 2       $print_lit (' avl - device available for use');
527      0615 2     IF .dev [dev$v_mnt]
528      0616 2       $print_lit (' mnt - device is mounted');
529      0617 2     IF .dev [dev$v_mbx]
530      0618 2       $print_lit (' mbx - device is a mailbox');
```

```

: 548      0636 2   IF .dev [dev$v_dmt]
549      0637 2   THEN
550      0638 2     $print_lit (' dmt - device marked for dismount');
551      0639 2   IF .dev [dev$v_elg]
552      0640 2   THEN
553      0641 2     $print_lit (' elg - device has error logging enabled');
554      0642 2   IF .dev [dev$v_all]
555      0643 2   THEN
556      0644 2     $print_lit (' all - device is allocated');
557      0645 2   IF .dev [dev$v_for]
558      0646 2   THEN
559      0647 2     $print_lit (' for - device is mounted foreign (i.e., non-file structured)');
560      0648 2   IF .dev [dev$v_sw1]
561      0649 2   THEN
562      0650 2     $print_lit (' sw1 - device is software write locked');
563      0651 2   IF .dev [dev$v_idv]
564      0652 2   THEN
565      0653 2     $print_lit (' idv - device capable of providing input');
566      0654 2   IF .dev [dev$v_odv]
567      0655 2   THEN
568      0656 2     $print_lit (' odv - device capable of providing output');
569      0657 2   IF .dev [dev$v_rnd]
570      0658 2   THEN
571      0659 2     $print_lit (' rnd - device allows random access');
572      0660 2   IF .dev [dev$v_rtm]
573      0661 2   THEN
574      0662 2     $print_lit (' rtm - device is realtime in nature');
575      0663 2   IF .dev [dev$v_rck]
576      0664 2   THEN
577      0665 2     $print_lit (' rck - device has read checking enabled');
578      0666 2   IF .dev [dev$v_wck]
579      0667 2   THEN
580      0668 2     $print_lit (' wck - device has write checking enabled');
581
582      0669 2
583      0670 2 RETURN 1;
: 583      0671 1 END;

```

.PSECT EXCH\$DBG_PLIT,NOWRT,2

69 20 65 63 69 76 65 64 20 2D 20 63 65 72 09 00BDC P.AGS:	.ASCII <9>\rec - device is record oriented\	:
74 6E 65 69 72 6F 20 64 72 6F 63 65 72 20 73 00BEB		
64 65 00BFA		
010E0020 00BFC P.AGR:	.LONG 17694752	
00000000 00C00 P.AGU:	.ADDRESS P.AGS	
65 67 61 69 72 72 61 63 20 2D 20 6C 63 63 09 00C04 P.AGU:	.ASCII <9>\ccl - carriage control device\<0>	
65 63 69 76 65 64 20 6C 6F 72 74 6E 6F 63 20 00C13		
00 00C22		
00 00C23	.ASCII <0>	
010E001E 00C24 P.AGT:	.LONG 17694750	
00000000 00C28 P.AGW:	.ADDRESS P.AGU	
69 20 65 63 69 76 65 64 20 2D 20 6D 72 74 09 00C2C P.AGW:	.ASCII <9>\trm - device is a terminal\<0>	
00 6C 61 6E 69 6D 72 65 74 20 61 20 73 00C3B		
010E001B 00C48 P.AGV:	.LONG 17694747	
00000000 00C4C P.AGY:	.ADDRESS P.AGW	
69 20 65 63 69 76 65 64 20 2D 20 72 69 64 09 00C50 P.AGY:	.ASCII <9>\dir - device is directory structur\	

EXCH\$DBG	Conditionally assembled debugging routines	16-Sep-1984 00:45:26	VAX-11 Bliss-32 V4.0-742
V04-000	exch\$dbg_utl_print_devchar	14-Sep-1984 12:29:02	[EXCHNG.SRC]EXCDBG.B32;1
73 75 20 72 6F 66 20 65 6C 62 61 6C 69 61 76 00E4B			
	00 65 00E5A	P.AHT: .LONG 17694751	
69 20 65 63 69 76 65 64 20 2D 20 74 6E 6D 09 00E60	010E001F 00000000	.ADDRESS P.AHU	
69 20 65 63 69 76 65 64 20 2D 20 74 6E 6D 09 00E64	00000000 00E60	P.AHW: .ASCII <9>\mnt - device is mounted\	
69 20 65 63 69 76 65 64 20 2D 20 78 62 6D 09 00E73	010E0018 00000000	00E7C P.AHV: .LONG 17694744	
69 20 65 63 69 76 65 64 20 2D 20 78 62 6D 09 00E80	00000000 00E80	.ADDRESS P.AHW	
00 00 78 6F 62 6C 64 20 2D 20 61 6D 20 73 61 20 73 00E84	010E001A 00000000	P.AHY: .ASCII <9>\mbx - device is a mailbox\<0><0>	
00E93 00E93	00000000 00EA0	P.AHX: .LONG 17694746	
6D 20 65 63 69 76 65 64 20 2D 20 74 6D 64 09 00EA4	00000000 00EB7	.ADDRESS P.AHY	
6F 6D 73 69 64 20 72 6F 66 20 2D 20 64 65 6B 72 61 00EC6	00000000 00EC6	P.AIA: .ASCII <9>\dmt - device marked for dismount\<0>	
	00 74 6E 75 00ECA		
	010E0021 00ED0	P.AHZ: .ASCII <0><0>	
	00000000 00ECC	.LONG 17694753	
68 20 65 63 69 76 65 64 20 2D 20 72 67 6C 65 09 00ED4	00000000 00EE3	.ADDRESS P.AIA	
6E 69 67 67 6F 6C 20 72 6F 72 2D 20 61 6E 65 20 67 00EF2	010E0027 00EF7	P.AIC: .ASCII <9>\elg - device has error logging ena\	
	00000000 00EFC	.ASCII \bled\<0>	
	00F00 00F04	.LONG 17694759	
69 20 65 63 69 76 65 64 20 2D 20 6C 6C 61 09 00F13	00000000 00F04	.ADDRESS P.AIC	
00 00 64 6F 6C 61 20 73 00F20	010E001A 00F20	P.AIE: .ASCII <9>\all - device is allocated\<0><0>	
	00000000 00F24	.LONG 17694746	
69 20 65 63 69 76 65 64 20 2D 20 72 6F 66 09 00F28	00000000 00F37	.ADDRESS P.AIE	
69 65 72 6F 66 20 64 65 74 6E 75 69 28 20 6E 67 00F46	010E001A 00F46	P.AIG: .ASCII <9>\for - device is mounted foreign (i\	
73 20 65 6C 69 66 2D 6E 6F 6E 20 2C 2E 65 2E 00F4B	00000000 00F5A	.ASCII \.e., non-file structured)\	
29 64 65 72 75 74 63 75 72 74 010E003C 00F64	010E003C 00F64	P.AIF: .LONG 17694780	
	00000000 00F68	.ADDRESS P.AIG	
69 20 65 63 69 76 65 64 20 2D 20 6C 77 73 09 00F6C	00000000 00F7B	P.AII: .ASCII <9>\swl - device is software write loc\	
74 69 72 77 20 65 72 61 77 74 66 6F 73 20 73 00F8A	00000000 00F8A	.ASCII \ked\<0><0>	
	00 00 64 6F 6C 20 65 00F8F	.LONG 17694758	
	010E0026 00F94	.ADDRESS P.AII	
63 20 65 63 69 76 65 64 20 2D 20 76 64 69 09 00F9C	00000000 00F98	P.AIH: .ASCII <9>\idv - device capable of providing \	
69 76 6F 72 70 20 66 6F 20 2D 65 6C 62 61 70 61 00FAB	010E0028 00F9C	.ASCII \input\	
	00000000 00FBA	.LONG 17694760	
74 75 70 6E 69 74 75 70 70 6E 69 00FBF	00000000 00FC4	.ADDRESS P.AIK	
	010E0028 00FC4	P.AIM: .ASCII <9>\odv - device capable of providing \	
63 20 65 63 69 76 65 64 20 2D 20 76 64 6F 09 00FCC	00000000 00FC8	.ASCII \output\<0><0><0>	
69 76 6F 72 70 20 66 6F 20 2D 65 6C 62 61 70 61 00FDB	010E0029 00FF8	.LONG 17694761	
	00000000 00FEA	.ADDRESS P.AIM	
00 00 00 74 75 70 74 75 6F 00FFC	010E0029 00FFC	P.AIL: .ASCII \output\<0><0><0>	
	00000000 01000	.LONG 17694761	
61 20 65 63 69 76 65 64 20 2D 20 73 64 6E 72 09 0100F	01000 0100F	.ADDRESS P.AIL	
63 61 20 6D 6F 64 6E 61 72 2D 20 73 77 6F 6C 6C 0101E	0100F 01023	P.AIO: .ASCII <9>\rnd - device allows random access\<0>	
	00 01023	.ASCII <0>	

69 20 65 63 69 76 65 64 20 2D 20 6D 61 74 72 09 010E0022 00000000 01024 P.AIN: .LONG 17694754	.ADDRESS P.AIO
6E 20 6E 69 20 65 6D 69 74 6C 61 65 72 20 73 0103B 0102C P.AIQ: .ASCII <9>\rtm - device is realtime in nature\	
69 20 65 63 69 76 65 64 20 2D 20 6B 63 72 09 010E0023 00000000 01050 P.AIP: .LONG 17694755	.ADDRESS P.AIQ
6E 69 6B 63 65 68 63 20 64 61 65 72 20 73 61 01067 01054 P.AIS: .ASCII <9>\rck - device has read checking ena\	
68 20 65 63 69 76 65 64 20 2D 20 6B 63 77 09 010E0027 00000000 01080 P.AIR: .LONG 17694759	.ADDRESS P.AIS
69 6B 63 65 68 63 20 65 74 69 72 77 20 73 61 01097 01084 P.AIU: .ASCII <9>\wck - device has write checking en\	
69 20 65 63 69 76 65 64 20 2D 20 6B 63 77 09 010E0028 00000000 01080 PAIT: .ASCII \abled\	.LONG 17694760
69 6B 63 65 68 63 20 65 74 69 72 77 20 73 61 010A6 010AB P.AIT: .LONG 17694760	.ADDRESS P.AIU

.PSECT EXCH\$DBG_CODE,NOWRT,2

52 00000000G 0004 00000	.ENTRY EXCH\$DBG_UTL_PRINT_DEVCHAR, Save R2	0579
07 04 AC 9E 00002	MOVAB LIB\$PUT_OUTPUT, R2	
0000' CF E9 00009	BLBC DEV, 1\$	0585
0000' CF 9F 0000D	PUSHAB P.AGR	0587
07 04 62 01 FB 00011	CALLS #1, LIB\$PUT_OUTPUT	
0000' CF 9F 00019 1\$:	BBC #1, DEV, 2\$	0588
07 04 62 01 FB 0001D	PUSHAB P.AGT	0590
0000' CF 9F 00025 2\$:	CALLS #1, LIB\$PUT_OUTPUT	
07 04 62 01 FB 00029	BBC #2, DEV, 3\$	0591
0000' CF 9F 00031 3\$:	PUSHAB P.AGV	0593
07 04 62 01 FB 0002C	CALLS #1, LIB\$PUT_OUTPUT	
0000' CF 9F 00035 4\$:	BBC #3, DEV, 4\$	0594
07 04 62 01 FB 00038	PUSHAB P.AGX	0596
0000' CF 9F 0003D 4\$:	CALLS #1, LIB\$PUT_OUTPUT	
07 04 62 01 FB 00041	BBC #4, DEV, 5\$	0597
0000' CF 9F 00049 5\$:	PUSHAB P.AGZ	0599
07 04 62 01 FB 0004D	CALLS #1, LIB\$PUT_OUTPUT	
0000' CF 9F 00051 6\$:	BBC #5, DEV, 6\$	0600
07 04 62 01 FB 00050	PUSHAB P.AHB	0602
0000' CF 9F 00055 6\$:	CALLS #1, LIB\$PUT_OUTPUT	
62 01 FB 00059 7\$:	BBC #6, DEV, 7\$	0603
04 AC 95 0005C 7\$:	PUSHAB P.AHD	0605
07 18 0005F TSTB DEV	CALLS #1, LIB\$PUT_OUTPUT	
0000' CF 9F 00061 BGEQ 8\$	PUSHAB P.AHF	0606
62 01 FB 00065 CALLS #1, LIB\$PUT_OUTPUT		
07 05 AC E9 00068 8\$:	BLBC DEV+1, 9\$	0608
0000' CF 9F 0006C PUSHAB P.AHH		
07 05 62 01 FB 00070 CALLS #1, LIB\$PUT_OUTPUT		
0000' CF 9F 00073 9\$:	BBC #5, DEV+1, T0\$	0610
05 E1 00073 9\$:	PUSHAB P.AHJ	0612
0000' CF 9F 00078		0614

07	05	62		01	FB 0007C	10\$:	CALLS #1, LIB\$PUT_OUTPUT		0615
			0000'	06	E1 0007F		BBC #6, DEV+1, T1\$		0617
		62		CF 9F 00084		PUSHAB P.AHL			
			05	01	FB 00088	11\$:	CALLS #1, LIB\$PUT_OUTPUT		0618
				07	95 0008B		TSTB DEV+1		
			0000'	CF 9F 00090		BGEQ 12\$			
		62		01	FB 00094		PUSHAB P.AHN		0620
			07	AC 9E 00097	12\$:	CALLS #1, LIB\$PUT_OUTPUT			
			0000'	CF 9F 0009B		BLBC DEV+2, 13\$			
		62		01	FB 0009F		PUSHAB P.AHP		0621
	07	06	AC	01	E1 000A2	13\$:	CALLS #1, LIB\$PUT_OUTPUT		0623
			0000'	CF 9F 000A7		BBC #1, DEV+2, T4\$			
			01	FB 000AB		PUSHAB P.AHR			
	07	06	62	02	E1 000AE	14\$:	CALLS #1, LIB\$PUT_OUTPUT		0624
			0000'	CF 9F 000B3		BBC #2, DEV+2, T5\$			
			01	FB 000B7		PUSHAB P.AHT			
	07	06	AC	03	E1 000BA	15\$:	CALLS #1, LIB\$PUT_OUTPUT		0627
			0000'	CF 9F 000BF		BBC #3, DEV+2, T6\$			
			01	FB 000C3		PUSHAB P.AHV			
	07	06	62	04	E1 000C6	16\$:	CALLS #1, LIB\$PUT_OUTPUT		0630
			0000'	CF 9F 000CB		BBC #4, DEV+2, T7\$			
			01	FB 000CF		PUSHAB P.AHX			
	07	06	AC	05	E1 000D2	17\$:	CALLS #1, LIB\$PUT_OUTPUT		0633
			0000'	CF 9F 000D7		BBC #5, DEV+2, T8\$			
			01	FB 000DB		PUSHAB P.AHZ			
	07	06	62	06	E1 000DE	18\$:	CALLS #1, LIB\$PUT_OUTPUT		0636
			0000'	CF 9F 000E3		BBC #6, DEV+2, T9\$			
			01	FB 000E7		PUSHAB P.AIB			
	07	06	AC	07	95 000EA	19\$:	CALLS #1, LIB\$PUT_OUTPUT		0641
			06	AC 95 000ED		TSTB DEV+2			
			0000'	CF 9F 000EF		BGEQ 20\$			
		62		01	FB 000F3		PUSHAB P.AID		0644
			07	AC E9 000F6	20\$:	CALLS #1, LIB\$PUT_OUTPUT			
			0000'	CF 9F 000FA		BLBC DEV+3, 21\$			
	07	07	62	01	FB 000FE		PUSHAB P.AIF		0645
			0000'	01	E1 00101	21\$:	CALLS #1, LIB\$PUT_OUTPUT		0650
				CF 9F 00106		BBC #1, DEV+3, 22\$			
			01	FB 0010A		PUSHAB P.AIH			
	07	07	AC	02	E1 0010D	22\$:	CALLS #1, LIB\$PUT_OUTPUT		0651
			0000'	CF 9F 00112		BBC #2, DEV+3, 23\$			
			01	FB 00116		PUSHAB P.AIJ			
	07	07	AC	03	E1 00119	23\$:	CALLS #1, LIB\$PUT_OUTPUT		0653
			0000'	CF 9F 0011E		BBC #3, DEV+3, 24\$			
			01	FB 00122		PUSHAB P.AIL			
	07	07	62	04	E1 00125	24\$:	CALLS #1, LIB\$PUT_OUTPUT		0654
			0000'	CF 9F 0012A		BBC #4, DEV+3, 25\$			
			01	FB 0012E		PUSHAB P.AIN			
	07	07	AC	05	E1 00131	25\$:	CALLS #1, LIB\$PUT_OUTPUT		0657
			0000'	CF 9F 00136		BBC #5, DEV+3, 26\$			
			01	FB 0013A		PUSHAB P.AIP			
	07	07	62	06	E1 0013D	26\$:	CALLS #1, LIB\$PUT_OUTPUT		0660
			0000'	CF 9F 00142		BBC #6, DEV+3, 27\$			
			01	FB 00146		PUSHAB P.AIR			
	07	07	AC	07	95 00149	27\$:	CALLS #1, LIB\$PUT_OUTPUT		0663
			0000'	AC 95 0014C		TSTB DEV+3			
			01	FB 0014E		BGEQ 28\$			
			07	AC 9F 0014E		PUSHAB PAIT			

EXCH\$DBG
V04-000

Conditionally assembled debugging routines
exch\$dbg_utl_print_devchar

M 7
16-Sep-1984 00:45:26 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:29:02 [EXCHNG.SRC]EXCDBG.B32;1

Page 39
(9)

62 01 FB 00152 CALLS #1, LIB\$PUT_OUTPUT
50 01 D0 00155 28\$: MOVL #1, R0
 04 00158 RET

: 0670
: 0671

: Routine Size: 345 bytes, Routine Base: EXCH\$DBG_CODE + 093D

EXCH\$DBG
V04-000

Conditionally assembled debugging routines
exch\$dbg_utl_print_devchar

N 7
16-Sep-1984 00:45:26
14-Sep-1984 12:29:02
VAX-11 Bliss-32 V4.0-742
[EXCHNG.SRC]EXCDBG.B32;1

Page 40
(10)

: 585 0672 1 END
: 586 0673 0 ELUDOM

.EXTRN LIB\$SIGNAL, LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
EXCH\$DBG_PLIT	4280 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)	
EXCH\$DBG_CODE	2710 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)	
EXCH\$DBG_OWN	180 NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)	

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	106	0	1000	00:01.9
\$255\$DUA28:[EXCHNG.OBJ]EXCLIB.L32;1	1151	56	4	79	00:01.4

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:EXCDBG/OBJ=OBJ\$:EXCDBG MSRC\$:EXCDBG/UPDATE=(ENH\$:EXCDBG)

: Size: 2710 code + 4460 data bytes
: Run Time: 00:53.1
: Elapsed Time: 02:55.9
: Lines/CPU Min: 760
: Lexemes/CPU-Min: 26418
: Memory Used: 348 pages
: Compilation Complete

0160 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

EXCDEFS
LTS

EXCDIRE
LTS

EXCDOS11
LTS

EXCDBG
LTS